(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : **0 614 317 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number : **94301252.6**

(51) Int. Cl.^{5} : **H04N 7/13**

(22) Date of filing : **22.02.94**

(30) Priority : **05.03.93 JP 45112/93**

(43) Date of publication of application :
07.09.94 Bulletin 94/36

(84) Designated Contracting States :
DE FR GB IT NL

(71) Applicant : **SONY CORPORATION**
7-35 Kitashinagawa 6-chome
Shinagawa-ku
Tokyo 141 (JP)

(72) Inventor : **Koyanagi, Hideki, c/o Intellectual Property Div.**
Sony Corporation,
6-7-35 Kitashinagawa
Shinagawa-ku, Tokyo 141 (JP)
Inventor : **Sumihiro, Hiroshi, c/o Intellectual Property Div.**
Sony Corporation,
6-7-35 Kitashinagawa
Shinagawa-ku, Tokyo 141 (JP)
Inventor : **Emoto, Seiichi, c/o Intellectual Property Div.**
Sony Corporation,
6-7-35 Kitashinagawa
Shinagawa-ku, Tokyo 141 (JP)
Inventor : **Wada, Tohru, c/o Intellectual Property Div.**
Sony Corporation,
6-7-35 Kitashinagawa
Shinagawa-ku, Tokyo 141 (JP)

(74) Representative : **Robinson, Nigel Alexander Julian et al**
D. Young & Co.,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) Video signal decoding.

(57) A digital video signal that has been encoded using motion- compensated prediction, transform encoding, and variable-length coding, is decoded using parallel processing. Frames of the video signal are divided into slices (1, 2, 3, 4) made up of a sequence of macroblocks (MB). The signal to be decoded is slice-wise divided for parallel variable-length decoding. Each variable-length-decoded macroblock is divided into its constituent blocks for parallel inverse transform processing. Resulting blocks of difference data are added in parallel to corresponding blocks of reference data. The blocks of reference data corresponding to each macroblock are read out in parallel from reference data memories (44, 45, 46, 47) on the basis of a motion vector (83) associated with the macroblock. Reference data corresponding to each macroblock is distributed for storage among a number of reference data memories.

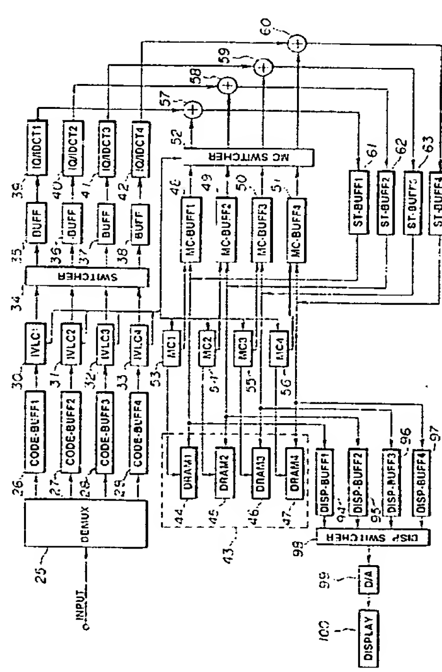

FIG.1

EP 0 614 317 A2

This invention relates to decoding of prediction-coded video signals, and more particularly is directed to the application of parallel processing to such decoding.

It is known to perform compression coding on video data which represents a moving picture in order to reduce the quantity of data to be recorded and/or transmitted. Such data compression may be useful, for example, in recording/reproducing systems using recording media such as magnetic tape or optical disks, and is also useful in transmission systems such as those used for video teleconferencing, video telephones, television broadcasting (including direct satellite broadcast), and the like. For example, it has been proposed by the Moving Picture Experts Group (MPEG) to compression-code moving picture video data utilizing motion-compensated prediction, transform processing using an orthogonal transformation such as the discrete cosine transform (DCT), and variable-length coding. A system for decoding and reproducing such compression-coded video data is illustrated in block diagram form in Figure 14 of the accompanying drawings.

As shown in Figure 14, a sequence of compression-coded video data is provided at an input terminal 101 for processing, in turn, by an inverse VLC (variable-length coding) circuit 102, an inverse quantization circuit 103, and an inverse DCT circuit 104. An adding circuit 105 forms a reconstructed frame of video data on the basis of a difference signal provided from the inverse DCT circuit 104 and predictive picture data (reference data) provided from a motion compensation circuit 106. The resulting reconstructed video data is stored in a frame memory 107.

The motion compensation circuit 106 forms the predictive picture data from reconstructed data previously stored in frame 107 on the basis of motion compensation information (including, for example, motion vectors) extracted from the input signal and supplied to the motion compensation circuit 106 by the inverse VLC circuit 102. Alternatively, with respect to frames for which predictive coding was not performed, such as "intra-frame" coded data, the motion compensation circuit 106 simply provides the value "0" to the adder 105. Reconstructed frames of video data are output from the frame memory 107 via a digital-to-analog converter 108 for display by a display device 109.

As the number of pixels in each frame of the video signal has increased from, for example, the 352 x 240 frame used for video telephones to the 720 x 480 frame used in the NTSC format or the 1920 x 1024 frame in a HDTV (high definition television) system, it was found to be difficult to perform the necessary processing using only one processor and one program execution sequence. For this reason, it has been proposed to divide each frame of the video data into a plurality of subframes, as illustrated in Figure 16 of the accompanying drawings, and then to provide a respective processor for each of the plurality of subframes, so that coding and decoding are performed with parallel processing by the plurality of processors. For example, Figure 15 of the accompanying drawings is a block diagram of a decoding system provided in accordance with this proposal.

In the system of Figure 15, input sequences of encoded video data, each representing a respective subframe, are respectively provided via input terminals 110-113 to processors (decoder blocks) 114-117. The processors 114-117 decode the respective data sequences based upon data supplied from frame memories 119-122, which store respective subframes and are assigned to respective ones of the processors 114-117. For example, processor 114 stores a subframe of decoded data in the memory 119. In order to provide motion compensation, a switching logic circuit 118 provided between the processors 114-177 and the frame memories 119-122, permits the processor 114 to read out data from an adjacent portion of the frame memory 120 as well as from all of frame memory 119. The switching logic circuit 118 also provides frames of output video data from the memories 119-120, via a digital-to-analog converter 123 for display on a display device 124.

The four data sequences respectively provided to the processors 114-117 can, for practical purposes, be combined into a single data sequence by providing headers for controlling multiplexing of the data sequence. For this purpose, a separation block (not shown) is provided upstream from the decoder for separating the combined data sequence into the four sequences to be provided to the respective processors. Examples of parallel processing techniques which use division of a video frame into subframes are disclosed in U.S. Patent No. 5,138,447 and Japanese Patent Application Laid Open No. 139986/1992 (Tokkaihei 4-139986).

As just described, according to the conventional approach, the video frame was generally divided into subframes which were processed in parallel by respective processors. However, when a frame is divided in this manner, there are restrictions on the extent to which the processors can access data that is outside of the processor's respective subframe. Although, as indicated above, a processor can access a region that adjoins its respective subframe, the extent of such access is limited in order to keep the scale of the switching logic circuit 118 from becoming unduly large. As a result, the degree of compression efficiency is reduced, and there are variations in the quality of the reproduced picture at the boundary between the subframes, which may result in visible artifacts at the subframe boundary.

In addition, the processing for compression-coding is carried out completely separately for each of the subframes, which makes it impossible to provide

compression-coding on the basis of data blocks in other subframes, a limitation that is not present when the frame is not divided into subframes. Accordingly, the compression coding method must be changed to accommodate the division into subframes, resulting in a lack of compatibility and a loss in compression efficiency.

Furthermore, if header data is added to the data sequence to be recorded or transmitted in order to provide for multiplexing the data sequence into the respective sequences provided to the parallel processors, the additional header data increases the overhead in the recorded data with a corresponding loss of efficiency, and it may also be necessary to change the coding procedure, and so forth.

In accordance with a first aspect of the present invention, there is provided an apparatus for decoding a coded video signal that represents an image frame, said coded video signal having been divided into a plurality of slices each of said slices being a sequence of macroblocks, each of said macroblocks being a two-dimensional array of picture elements of said image frame, said coded video signal being a bit stream that represents a sequence of said slices which together represent said image frame, said bit stream including a plurality of synchronizing code signals, each of which is associated with a respective one of said slices for indicating a beginning of the respective slice, the apparatus comprising:

a plurality of decoding means each for decoding a respective portion of said coded video signal that represents said image frame; and

distributing means responsive to said synchronizing code signals for distributing said slices among said plurality of decoding means.

According to a second aspect of the invention, there is provided an apparatus for decoding input signal blocks that were formed by transform encoding and then variable-length encoding blocks of video data, the apparatus comprising:

decoding means for variable-length decoding a series of said input signal blocks;

parallel data means for forming plural parallel data streams, each of which includes respective ones of said series of input signal blocks which were variable-length decoded by said decoding means; and

a plurality of inverse transform means each for receiving a respective one of said parallel data streams and for performing inverse transform processing on the variable-length decoded signal blocks in the respective data stream.

In preferred embodiments of the apparatus just described, the decoding circuit is one of a plurality of decoding circuits for variable-length decoding respective series of input signal blocks, and the apparatus further includes a distributing circuit for forming the respective series of input signal blocks to be decoded by the plural decoding circuits from a bit stream representing an image frame, and the respective series of input signal blocks are formed in response to synchronizing signals provided at predetermined intervals in the bit stream representing the image frame.

According to a third aspect of the invention, there is provided an an apparatus for decoding an input digital video signal which includes groups of blocks of prediction-coded difference data, each of said groups consisting of a predetermined plurality of said blocks and having a respective motion vector associated therewith, each of said blocks of prediction-coded difference data having been formed on the basis of the respective motion vector associated with the respective group which includes said block, the apparatus comprising:

output means for supplying in parallel blocks of prediction-coded difference data contained in one of said groups of blocks;

reference data means for supplying in parallel plural blocks of reference data, each of said blocks of reference data being formed on the basis of the motion vector associated with said one of said groups of blocks and corresponding to one of said blocks of prediction-coded difference data supplied by said output means; and

a plurality of adding means each connected to said output means and said reference data means for adding a respective one of said blocks of prediction-coded difference data and the corresponding block of reference data.

In preferred embodiments of the invention, the reference data circuit includes a plurality of reference data memories from which reference data is read out in parallel on the basis of the motion vector associated with that group of blocks, a plurality of buffer memories for temporarily storing reference data read out from the plurality of reference data memories and a distribution circuit. According to one alternative embodiment of this aspect of the invention, each of the buffer memories is associated with a respective one of the reference data memories and is controlled on the basis of the motion vector for reading out the reference data temporarily stored therein, and the distributing circuit is connected between the buffer memories and the adding circuits and distributes the reference data stored in the buffer memories among the adding circuits on the basis of the motion vector. According to another alternative embodiment of this aspect of the invention, each of the buffer memories is associated with one of the adding circuits and the distributing circuit is connected between the reference data memories and the buffer memories for distributing among the buffer memories, on the basis of the motion vector associated with that group of blocks, the reference data read out from the reference data memories.

According to a fourth aspect of the invention,

there is provided a method of decoding a coded video signal that represents an image frame, said coded video signal having been divided into a plurality of slices each of said slices being a sequence of macroblocks, each of said macroblocks being a two-dimensional array of picture elements of said image frame, said coded video signal being a bit stream that represents a sequence of said slices which together represent said image frame, said bit stream including a plurality of synchronizing code signals, each of which is associated with a respective one of said slices for indicating a beginning of the respective slice, the method comprising the steps of:

providing a plurality of decoding means each for decoding a respective portion of said coded signal that represents said video frame; and

distributing said slices among said plurality of decoding means in response to said synchronizing code signals.

The data representing each macroblock may be distributed block-by-block among the plurality of memories or line-by-line in a cyclical fashion among the plurality of memories.

A video signal decoding apparatus may be provided in which the input coded signal is distributed for parallel processing among several decoding circuits on the basis of synchronizing code signals that are provided in the signal in accordance with a conventional coding standard. In this way, parallel decoding can be precisely carried out on the basis of synchronizing signals provided in accordance with a conventional coding method and during time periods available between the synchronizing signals. In this way, restrictions on the conventional coding method can be reduced.

In addition, the data may be sequenced on the basis of "slices" which are a standard subdivision of a video frame constituting a plurality of macroblocks and the slices of data are distributed among decoding circuits so that high speed parallel decoding may be carried out.

Further, each of the blocks making up a macroblock may be distributed to a respective inverse transformation circuit so that inverse transform processing can be carried out simultaneously in parallel for all of the blocks of a macroblock, and the inverse transform blocks are then combined, in parallel, with reference data to recover the video signal which had been predictive-coded. The reference data, in turn, may be provided from parallel memories at the same time on the basis of the motion compensation vector for the particular macroblock, and in such a way that there is no need to place restrictions on the motion-compensation carried out during the predictive coding. For example, there is no need to limit the range of the motion vector.

Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 is a block diagram of an embodiment of an apparatus for decoding a moving picture video data signal;

Figure 2 is a schematic illustration of a manner in which video data corresponding to an image frame is distributed for decoding;

Figure 3 is a timing diagram which illustrates operation of a buffer memory provided in the apparatus of Figure 1;

Figure 4 is a block diagram which illustrates a code buffering arrangement provided upstream from variable-length decoder circuits provided in the apparatus of Figure 1;

Figure 5 is a timing diagram which illustrates operation of the code buffering arrangement shown in Figure 4;

Figure 6 is a block diagram which shows an alternative code buffering arrangement provided upstream from variable-length decoder circuits provided in the apparatus of Figure 1;

Figure 7 is a timing diagram which illustrates operation of the code buffering arrangement shown in Figure 6;

Figures 8(A), 8(B) and 8(C) together schematically illustrate a manner in which reference data is provided on the basis of a motion vector to adders that are part of the apparatus of Figure 1;

Figure 9 is a timing diagram which illustrates an operation for providing reference data to the adders which are part of the apparatus of Figure 1;

Figure 10 is a block diagram of another embodiment of an apparatus for decoding a moving picture video data signal;

Figure 11(A), 11(B) and 11(C) together schematically illustrate a manner in which reference data is provided on the basis of a motion vector to adders that are part of the apparatus of Figure 10;

Figures 12(A) and 12(B) together schematically illustrate an alternative manner in which reference data is provided on the basis of a motion vector to the adders which are part of the apparatus of Figure 10;

Figure 13 is timing diagram which illustrates an operation for providing reference data according to the example shown in Figure 12;

Figure 14 is a block diagram of a conventional apparatus for decoding and reproducing a moving picture video data signal;

Figure 15 is a block diagram of a portion of a conventional apparatus for decoding and reproducing a moving picture video data signal by means of parallel processing; and

Figure 16 schematically illustrates operation of the conventional decoding apparatus of Figure 15.

A preferred embodiment of the invention will now be described, initially with reference to Figure 1.

Figure 1 illustrates in block diagram form an apparatus for decoding a moving picture video data signal that has been coded according to a proposed MPEG standard system.

An input bit stream representing the coded video data signal is provided to a demultiplexer 25, by means of which the input signal is distributed, slice-by-slice, to code buffers 26-29.

Figure 2 illustrates the slice-by-slice distribution of the input data. As is well known to those who are skilled in the art, each slice is a sequence of macroblocks transmitted in raster scanning order. The starting point of each slice is indicated by a synchronizing code signal, and the slices are provided so that transmission errors and the like can be confined to a single slice, because after an error occurs, proper coding can resume at the synchronizing code signal provided at the beginning of the subsequent slice. Accordingly, the demultiplexer 25 is provided with a circuit which detects the synchronizing code signals, and distribution of the input signal among the code buffers 26-29 is carried out in response to the detected synchronizing code signals.

As is also well known, the motion vectors provided with respect to each macroblock, and the DC coefficients for each block, are differentially encoded. In other words, only the difference between respective motion vectors for the current macroblock and the preceding macroblock is encoded and transmitted, and also, only the difference between the respective DC coefficient for the present block and that of the preceding block are coded and transmitted.

As indicated in Figure 2, the first, fifth, ninth, etc. slices of each image frame are stored in the first code buffer 26, and these slices are provided for variable-length decoding by a variable-length decoder circuit 30. Similarly, the second, sixth, tenth, etc. slices of the image frame are stored in the second code buffer 27 for variable-length decoding by variable-length decoder circuit 31; the third, seventh, eleventh, etc. slices are stored in the third code buffer 28 for variable-length decoding by the variable-length decoder circuit 32; and the fourth, eighth, twelfth, etc. slices are stored in the fourth code buffer 29 for variable-length decoding by the variable-length decoder circuit 33.

According to the example shown in Figure 2, the number of macroblocks in each slice is fixed, so that it will not be necessary for any of the variable-length decoders to wait. As result, decoding carried on by the variable-length decoders is synchronized and is carried out efficiently.

It will be understood that, although the number of macroblocks per slice is fixed, the number of bits per slice in the input signal will vary because of variable-length encoding. Nevertheless, the number of macroblocks per slice output by each variable-length decoding circuit is the same according to this example.

In the example shown in Figure 2, each slice is shown as being one macroblock high and extending horizontally entirely across the image frame, so that each slice consists of one row of macroblocks. However, it is also within the contemplation of this invention to provide for slices having a fixed length in terms of macroblocks that is longer or shorter than one row of macroblocks. It is further contemplated that the number of macroblocks per slice may be variable within each frame and/or from frame to frame and that the positions of slices within a frame may vary. In case variable-length slices are provided within a frame, it will be appreciated that the number of macroblocks distributed to each of the variable-length decoders may be unbalanced, in which case some of the variable-length decoders may be required to output filler macroblocks (all zeros for example) until other decoders have "caught up". Furthermore, it is provided that variable-length decoding of slices from the next image frame will not proceed until all of the slices of the current frame have been variable-length decoded.

It will be recognized that any loss of decoding efficiency that results from the occasional need to interrupt the processing by some of the variable length decoders is compensated for by the fact that the coding can be performed with slices that have a variable length in terms of macroblocks.

Details of the variable-length decoding processing will now be described.

Data which has been decoded by the respective variable length decoders are transferred to buffer memories 35-38 by way of switcher 34. Figure 3 illustrates the manner in which data is distributed to, and output from, the buffer memories 35-38. It will be noted that, upstream from the buffers 35-38, processing had been performed in a slice-wise parallel manner, but downstream from the buffers 35-38 processing is performed in a block-wise parallel manner. In particular, the four blocks of luminance data making up a macroblock are output in parallel from respective ones of the buffer memories 35-38. (It will be understood that a macroblock also includes chrominance blocks. For example, in the 4:2:2 format, each macroblock includes four blocks of chrominance data in addition to the four blocks of luminance data. The discussion from this point forward will deal only with the luminance data blocks, it being understood that the corresponding four chrominance data blocks can be processed in a similar manner.)

Referring again to Figure 3, it will be seen that the variable length decoders 30-33 respectively output simultaneously the respective first block of the first through fourth slices. The respective first blocks are distributed among the buffer memories 35-38 so that the first block of the first slice (i.e., the first block of the first macroblock of the first slice) is stored in the first buffer memory 35, the second block of the first slice is stored in the second buffer memory 36, the

third block of the first slice is distributed to the third buffer memory 37, and the fourth block of the first slice is distributed to the fourth buffer memory 38. As a result, all four blocks of a single macroblock can be read out in parallel by the respective buffer memories 35-38, so that block-wise parallel processing can be accomplished downstream. Such processing includes conventional inverse transform processing in accordance with zig-zag scanning.

In the example just discussed, each buffer memory preferably has two banks which each have the capacity of storing four data blocks.

The block-wise parallel data provided from the buffer memories 35-38 is subjected to inverse quantization and inverse discrete cosine transform processing in parallel at processing blocks 39-42. Thereafter, motion compensation processing for the four blocks of the macroblock is also carried out in parallel. Reference picture data for each macroblock is extracted from previously reproduced (i.e., previously reconstructed) image data stored in a frame memory 43. The reference picture data is formed on the basis of the motion vector which corresponds to the macroblock being processed and is used to form decoded data in combination with difference data output from the processing blocks 39-42. In this example, since motion compensation processing is carried out in parallel for each macroblock (four blocks) of luminance data, the motion vectors provided to motion compensation processing blocks 53-56 from the variable length decoders 30-33 always correspond to each other at any given time. For this reason, an MC (motion compensation) switcher 52 is used to switch a data bus, so that it is possible to provide motion compensation processing of the reference data transferred to MC buffer memories 48-51 in such a manner that memory accessing by the motion compensation processing blocks 53-56 does not overlap. As a result, the motion compensation search range, and accordingly the permissible range of the motion vector, is not limited. Details of motion compensation processing will be provided below.

Reproduced decoded image data formed in parallel at adders 57-60 is stored via four parallel processing paths in the frame memory 43 by way of storage buffers 61-64. Moreover, sequences of images for which the reproduced (reconstructed) data is stored in memory 43 are output to a digital-to-analog converter 99 through display buffer memories 94-97, and a display switcher 98 which is switched according to appropriate display timing. The D/A converted signal is then displayed on a display device 100.

There will now be described, with reference to Figure 4, details of a buffering arrangement provided upstream from the variable length coders of the apparatus of Figure 1.

As shown in Figure 4, an input signal bit stream is received at an input terminal 65 and provided therefrom to a demultiplexer 66 which divides the bit stream at the beginning of each slice and distributes the slices among code buffer memories 67-70. The slices of data are output respectively from the code buffer memories 67-70 to variable-length decoders 71-74, and variable-length decoded data is respectively output from each of the variable-length decoders 71-74 via output terminals 75-78.

The buffering and decoding operations carried out by the circuitry shown in Figure 4 will now be described with reference to the timing diagram shown in Figure 5.

In particular, the input bit stream received at the terminal 65 is divided at the beginning of each slice by the demultiplexer 66. Because synchronizing code signals indicative of the beginning of each slice are included at intervals corresponding to a plural number of macroblocks (such intervals each being referred to as a slice), the synchronizing code signals are detected at the demultiplexer 65 for the purpose of performing the division of the bit stream into slices.

As shown in Figure 5, a sequence of the resulting slices are written in a cyclical fashion into the code buffer memories 67-70. In particular, slice 1, slice 5, slice 9, etc. are written into the code buffer memory 67; slice 2, slice 6, slice 10, etc. are written into the code buffer memory 68; slice 3, slice 7, slice 11, etc. are written into the code buffer memory 69; and slice 4, slice 8, slice 12, etc. are written into the code buffer memory 70.

At a point when slice 4 has been written into the code buffer memory 70, the slices 1-4 are respectively read out in parallel from the code buffer memories 67-70 to the four variable-length decoders 71-74 and variable-length decoding begins.

The variable-length decoders 71-74 each complete decoding processing of a macroblock from a respective slice within the same time. Decoded data produced by variable-length decoder 71 is output via terminal 75; decoded data produced by variable-length decoder 72 is output via terminal 76; decoded data produced by variable-length decoder 73 is output via terminal 77; and decoded data produced by variable-length decoder 74 is output via terminal 78. All of the decoded data is supplied to the switcher 34 (Figure 1). In addition, decoded motion vector data is provided from the variable-length decoders to the MC switcher 52 and motion compensation processing blocks 53-56.

It should be understood that, in Figure 5, the symbol "1-1" shown in the output of IVLC1 (variable-length decoder 71) is indicative of the first block of slice 1. Similarly, for example, "4-1" shown in the output of IVLC4 (variable-length decoder 74) is indicative of the first block of slice 4.

An alternative code buffering arrangement provided upstream from the variable-length decoders is shown in Figure 6.

In Figure 6, the input bit stream is again received at an input terminal 65 and provided therefrom to a demultiplexer 79, at which the bit stream is divided at the beginning of each slice. Immediately downstream from the demultiplexer 79 is a code buffer memory 80 which has respective regions in each of which a slice of data can be stored. Additional buffer memories 90-93 are provided downstream from the buffer memory 80. In a similar manner to the arrangement of Figure 4, the buffered data output from each of the buffer memories 90-93 is provided to a respective one of the variable-length decoders 71-74, and the decoded data output from the variable-length decoders 71-74 is provided at respective output terminals 75-78.

Operation of the code buffering arrangement shown in Figure 6 will now be described with reference to the timing diagram of Figure 7.

As before, the input bit stream provided from the terminal 65 is divided at the beginning of each slice by the demultiplexer 79 on the basis of synchronizing code signals provided at intervals corresponding to a number of macroblocks.

As shown in Fig. 7, respective slices are written in a cyclical fashion into the regions 1-4 of the buffer memory 80. In particular, slice 1, slice 5, slice 9, etc. are written into region 1; slice 2, slice 6, slice 10, etc. are written into region 2; slice 3, slice 7, slice 11, etc. are written into region 3; and slice 4, slice 8, slice 12, etc. are written into the region 4.

At a point when slice 4 has been written into region 4, the data stored in the four regions are sequentially read out from the code buffer memory 80. As a result, slices 1, 5, 9, etc. are read out from region 1 and written into buffer memory 90; slices 2, 6, 10, etc., are read out from region 2 and written into buffer memory 91; slices 3, 7, 11, etc. are read out from region 3 and written into buffer memory 92, and slices 4, 8, 12, etc. are read out from region 4 and written into buffer memory 93.

At a time when the contents of region 4 have been written into the buffer memory 93, the data respectively stored in the buffer memories 90-93 is read out in parallel to the variable-length decoders 71-74, and decoding processing starts at that time.

The variable-length decoders 71-74 each complete the decoding processing of a respective macroblock within the same time. Decoded data produced by variable length decoder 71 is output via terminal 75; decoded data produced by variable-length decoder 72 is output via terminal 76; decoded data produced by variable-length decoder 73 is output via terminal 77; and decoded data produced by variable-length decoder 74 is output via terminal 78. This decoded data is supplied to the switcher 34, and in addition, decoded motion vector data is supplied from the variable-length decoders to MC switcher 52 and to motion compensation processing blocks 53-56.

As was the case with Figure 5, in Figure 7 the symbol "1-1" is indicative of the first block in slice 1, which is decoded by variable-length decoder 71, while "4-1" is indicative of the first block of slice 4, which is decoded by the variable-length decoder 74.

With respect to the buffering arrangement shown in Figure 4, it is possible to use certain distribution methods with respect to input data streams which have a processing unit which is shorter than a slice and are included in a layer (known as an "upper layer") which has processing units which are longer than a slice. With respect to an input data stream which has such a format, it is possible to simultaneously write the upper layer into the code buffer memories 67-70 in order to provide parallel data to the variable-length decoders 71-74. Alternatively, the bit stream for the upper layer can be written into one of the four code buffer memories so that the upper layer is decoded by only one of the four variable-length decoders, with parameters being set at the other variable-length decoders. According to another possible method, an additional processor is provided to decode the upper layer bit stream so as to set parameters at the four variable-length decoders.

On the other hand, using the arrangement shown in Figure 6, the upper layer bit stream can be written into one of the four regions of the buffer memory 80 and the contents of that region can be simultaneously written into the buffer memories 90-93 for parallel processing by the variable-length decoders 71-74. According to an alternative method, the upper layer bit stream is written into one of the four regions of the buffer memory 80 so that the data is written into one of the four buffer memories 90-93 and is then decoded by one of the four variable-length decoders in order to set parameters at the other variable-length decoders.

According to another alternative method, a separate processor is provided to decode the upper layer bit stream in order to set parameters at the four variable-length decoders. As a further method, the demultiplexer 79 repeatedly writes the upper layer bit stream into the four regions of the buffer memory 80 so that the data is simultaneously written from each region into the buffer memories 90-93 for parallel processing in the variable-length decoders 71-74.

In these ways, distribution of the data stream, and parallel processing thereof, can be carried out on the basis of parameters included in the data stream.

Details of decoding processing with respect to motion-compensated predictive-coded data will now be described.

Figure 8(A) illustrates a manner in which reference image data is distributed among and stored in DRAMs 44-47 making up the frame memory 43. Each image frame is, as indicated above, divided into macroblocks, and each macroblock is formed of four blocks. Each of the four blocks is, in this particular example, an 8 x 8 array of pixel elements, and each of

the blocks constitutes one of four quadrants of its respective macroblock. The data with respect to each macroblock is divided among the four DRAMs 44-47. In particular, all of the first blocks (upper left blocks) of all of the macroblocks are stored in DRAM 44, all of the second blocks (upper right blocks) of all of the macroblocks are stored in DRAM 45, all of the third blocks (lower left blocks) of all of the macroblocks are stored in DRAM 46, and all of the fourth blocks (lower right blocks) of all of the macroblocks are stored in DRAM 47. Accordingly, it will be seen that the reference data is distributed among DRAMs 44-47 in a checkered pattern.

Continuing to refer to Figure 8(A), the square labelled 81 represents the geometric area of the image frame which corresponds to the macroblock which is currently being decoded (reconstructed), and reference numeral 82 represents the motion vector associated with that macroblock, according to the example shown in Figure 8(A). In addition, the reference numeral 83 represents the reference data stored in the DRAMs 44-47 and indicated by the motion vector 82 as corresponding to the current macroblock 81. The data represented by the shaded square 83 is read out from the DRAMs 44-47 under control of motion compensation processing blocks 53-56 on the basis of the motion vector 82. In particular, the data corresponding to the "DRAM1" portion of the square 83 (i.e., a central portion of the square 83) is read out from DRAM 44 to motion compensation buffer 48 under the control of motion compensation processing block 53. Similarly, the portions of the shaded square 83 which overlap with squares labelled "DRAM2" (i.e., central portions of the left and right sides of the square 83) are read out from DRAM 45 to motion compensation buffer 49 under control of motion compensation processing block 54. Also, the portions of the shaded square 83 which overlap the squares labelled "DRAM3" (i.e., the central portions of the upper and lower edges of the square 83) are read out from DRAM 46 to motion compensation buffer 50 under control of motion compensation processing block 55. Finally, the portion of the shaded square 83 which overlaps with squares labelled "DRAM4" (i.e., corner regions of the square 83) are read out from the DRAM 47 to motion compensation buffer 51 under control of motion compensation processing block 56.

Figure 8(B) is a schematic illustration of the reference data read out from the respective DRAMs 44-47 and stored in respective motion compensation buffers 48-51. This data stored in the four motion compensation buffers 48-51 represents the reference data for the macroblock which is currently to be reconstructed. However, the data as stored in the individual motion compensation buffers does not correspond to the data required for each of the adders 57-60. Therefore, the MC switcher 52 is provided between the motion compensation buffers 48-51 and the adders 57-60 so that the correct reference data is distributed from the motion compensation buffers to the adders. The reference data which is supplied to each of the adders 57-60 is schematically illustrated in Figure 8(C).

Figure 9 illustrates the timing, according to the example shown in Figure. 8(A), at which data read out from the motion compensation buffers 48-51 is routed among the adders 57-60.

The processing of the four blocks making up the macroblock proceeds, as indicated before, in parallel, with the respective first lines of each of the blocks being processed simultaneously, then the second lines, and so forth. With respect to the first lines of the blocks, initially, at a starting time tl (Figure 9), data from motion compensation buffer 51 is routed to adder 57, data from motion compensation buffer 50 is routed to adder 58, data from motion compensation buffer 49 is routed to adder 59, and data from motion compensation buffer 48 is routed to adder 60. At a changeover point in the processing of the first lines, indicated by time t2 in Figure 9, the routing is changed so that data from motion compensation buffer 50 is routed to adder 57, data from motion compensation buffer 51 is routed to adder 58, data from motion compensation buffer 48 is routed to adder 59, and data from motion compensation buffer 49 is routed to adder 60. This routing state continues until the end of the first line (indicated by time t3) and then the procedure that was followed for the first lines is carried out again with respect to the second lines. The same procedure is then continued through the nth lines, but upon completion of the nth lines of the block, as indicated at time t4, a different routing pattern is established for the beginning of the (n + 1)th lines. According to this pattern, data from motion compensation buffer 49 is provided to adder 57, data from motion compensation buffer 48 is provided to adder 58, data from motion compensation buffer 51 is provided to adder 59, and data from motion compensation buffer 50 is provided to adder 60. This routing arrangement continues until a changeover point in the (n + 1)th lines, indicated by time t5, at which the routing arrangement is changed so that data from motion compensation buffer 48 is routed to adder 57, data from motion compensation buffer 49 is routed to adder 58, data from motion compensation buffer 50 is routed to adder 59, and data from motion compensation buffer 51 is routed to adder 60. On the completion of the process for the (n + 1)th line (indicated by time t6), the procedure carried out for the (n + 1)th lines is repeated with respect to each of the remaining lines of the blocks until the last (eighth) lines have been processed, at which point (indicated by time t7) processing for the macroblock is complete. Processing for the next macroblock then begins, on the basis of the motion vector associated with the next macroblock.

It will be appreciated that the reference data sup-

plied to the adders 50-60 is added by the adders to the current difference data supplied thereto from the processing circuits 39-42 so that macroblocks of reconstructed image data are produced. It will also be recognized that the storage of the reference data according to the above-described checkered pattern in the frame memory 43, and the above-described method of reading out, buffering, and switching the reference data makes it possible to provide motion-compensation decoding processing without any restriction on the range of the motion vector, and in such a manner that memory accesses do not overlap.

In the embodiment illustrated in Figure 1, the MC switcher 52 is provided between the motion compensation buffers 48-51 and the adders 57-60. However, according to an alternative embodiment, shown in Figure 10, the MC switcher 52 can be provided between the DRAMs 44-47 and the motion compensation buffers 48-51, with each of the buffers 48-51 connected directly to, and providing data exclusively to, a respective one of the adders 57-60.

A method of operating the embodiment illustrated in Figure 10 will be described with reference to Figures 11(A)-(C).

Figure 11(A) is similar to Figure 8(A), and shows a square 84 which represents the geometric area corresponding to the macroblock currently being processed, motion vector 85 associated with the current macroblock, and a shaded square 86 which represents the appropriate reference data for the current macroblock as indicated by the motion vector 85. It will also be noted that the reference data is distributed for storage among the DRAMs 44-47 in a block-wise manner according to the same checkered pattern shown in Figure 8(A).

Under control of the motion compensation processing blocks 53-56, and on the basis of the motion vector for the current macroblock, data is read out from the DRAMs 44-47 and routed to the motion compensation buffers 48-51 by the MC switcher 52 so that all of the reference data to be provided to the adder 57 is stored in the motion compensation buffer 48, all of the reference data to be provided to the adder 58 is stored in the motion compensation buffer 49, all of the reference data to be provided to the adder 59 is stored in the motion compensation buffer 50, and all of reference data to be provided to the adder 60 is stored in the motion compensation buffer 51. Referring to Figures 11(A) and (B), it will be noted that the data represented by the upper left quadrant of the shaded square 86 is stored in the motion compensation buffer 48, the data represented by the upper right quadrant of the shaded square 86 is stored in the motion compensation buffer 49, the data represented by the lower left quadrant of the shaded square 86 is stored in the motion compensation buffer 50, and the data represented by the lower right quadrant of the shaded square 86 is stored in the motion compensation buffer 51. More specifically, during an initial read out period, data is simultaneously read out from all four of the DRAMs 44-47 and routed such that data from a portion of the DRAM 47 is stored in motion compensation buffer 48, while data from a portion of DRAM 46 is stored in motion compensation buffer 49, data from a portion of DRAM 45 is stored in motion compensation buffer 50, and data from a portion of DRAM 44 is stored in motion compensation buffer 51. During a second read out period there is again simultaneous reading out of data from the four DRAMs, but now the routing is such that data from a portion of DRAM 46 is stored in motion compensation buffer 48, data from a portion of DRAM 47 is stored in motion compensation buffer 49, data from a portion of DRAM 44 is stored in motion compensation buffer 50, and data from a portion of DRAM 45 is stored in motion compensation buffer 51. Moreover, during a third read out period, again there is simultaneous read out from all of the DRAMs, but routing is performed so that data from a portion of DRAM 45 is stored in motion compensation buffer 48, data from a portion of DRAM 44 is stored in motion compensation buffer 49, data from a portion of DRAM 47 is stored in motion compensation buffer 50, and data from a portion of DRAM 46 is stored in motion compensation buffer 51. Then, during a final read out period, data is simultaneously read out from four DRAMs and routed such that data from a portion of DRAM 44 is stored in motion compensation buffer 48, data from a portion of DRAM 45 is stored in motion compensation buffer 49, data from a portion of DRAM 46 is stored in motion compensation buffer 50, and data from a portion of DRAM 47 is stored in motion compensation buffer 51.

It will be observed that data from every one of the four DRAMS is thus stored in each of the motion compensation buffers. Moreover, with reading of the data from the DRAMs and control of the MC switcher 52 on the basis of the motion vector for the current macroblock, memory access can be performed without overlap.

Also, because each of the motion compensation buffers are associated exclusively with a respective adder, and the reference data has been stored appropriately therein, as shown in Figure 11(C), there is also no difficulty in accessing the motion compensation buffers.

There will now be described, with reference to Figures 12 and 13, in addition to Figure 10, an alternative method of operating the embodiment of Figure 10 so that the appropriate reference data is stored in each of the motion compensation buffers 48-51.

As indicated in Figure 12(A), according to this alternative method of operation, the reference data is distributed line-by-line among the DRAMS 44-47, rather than block-by-block, as in the technique shown in Figure 11(A). For example, referring again to Figure 12(A), the data for the first line of each macroblock

(i.e., the first line of the first and second blocks of the macroblock), is stored in DRAM 44, the second line of data of each macroblock is stored in DRAM 45, the third line of data for each macroblock is stored in DRAM 46, the fourth line of each macroblock is stored in DRAM 47, the fifth line of each macroblock is stored in DRAM 44, and so forth, continuing in a cyclical fashion, line-by-line. It should be understood that the data for the ninth line of each macroblock (i.e., the first line of data in the third and fourth blocks of each macroblock) is stored in DRAM 44, whereas the data for the last line of each macroblock (i.e., the last line of the last two blocks of the macroblock) is stored in DRAM 47. Accordingly, the reference data is distributed among the DRAM 44-47 according to a striped pattern, rather than the checkered pattern of Figure 11(A).

In Figure 12(A), the square labelled 87 represents the geometric area which corresponds to the macroblock which is currently to be decoded, the motion vector 88 is the motion vector associated with the current macroblock, and the square 89 represents the appropriate reference data for the current macroblock, as indicated by the motion vector 88.

Figures 12(B) and Figure 13 indicate the sources of data and the timing according to which the appropriate reference data is stored in the motion compensation buffers 48-51. As before, data is read out from the DRAMS 44-47 and routed by MC switcher 52 under the control of the motion compensation processing blocks 43-56 and on the basis of the motion vector for the current macroblock.

In particular, during a first time slot, the reference data corresponding to the first line of the first block is read out from DRAM 47 and stored in motion compensation buffer 48. During the same time slot, reference data corresponding to the eighth line of the second block is read out from DRAM 46 and stored in motion compensation buffer 49. reference data for the seventh line of the third block is read out from DRAM 45 and stored in motion compensation buffer 50. and reference data for the sixth line of the fourth block is read out from DRAM 44 and stored in motion compensation buffer 51.

In the next (second) time slot, a one line shift in routing occurs, so that reference data for the second line of the first block is read out from DRAM 44 and stored in motion compensation buffer 48, reference data for the first line of the second block is read out from DRAM 47 and stored in motion compensation buffer 49, reference data for the eighth line of the third block is read out from DRAM 46 and stored in motion compensation buffer 50, and reference data for the seventh line of the fourth block is read out from DRAM 45 and stored in motion compensation buffer 51.

The one-line shifts are continued in each of the succeeding six time slots so that the data is read out, routed and stored in the motion compensation buffers according to the pattern shown in Figures 12(D) and 13. It will be observed that memory access occurs, as before, without overlapping.

As a result, the reference data which is to be supplied to adder 57 is stored in motion compensation buffer 48, reference data which is to be supplied to adder 58 is stored in motion compensation buffer 49, reference data which is to be supplied to adder 59 is stored in motion compensation buffer 50, and reference data which is to be supplied to adder 60 is stored in motion compensation buffer 51. Again, there is no problem with overlapping memory accesses with respect to the motion compensation buffers.

Although the above embodiments of the present invention have been described with respect to a decoding apparatus, it should be understood that the same could also be applied to a local decoder provided in a data encoding apparatus.

The moving picture video data decoding apparatus provided in accordance with this invention distributes an input data stream for parallel decoding processing on the basis of synchronizing code signals present in the data stream, and the decoding processing is continuously carried out within a time period between synchronizing codes. Accordingly, there is no limitation placed on the coding method with respect to time periods between synchronizing codes. Thus, parallel decoding processing can be carried out with respect to data that has been encoded by a conventional method, which difference-codes motion vectors, DC coefficients and the like on the basis of differences between a current block and a previous block.

In addition, in the decoding apparatus provided in accordance with this invention, the blocks making up a macroblock are simultaneously processed in parallel so that video data that has been encoded by a conventional encoding method, without modification, can be reproduced at high speed.

Furthermore, decoding of motion-compensation coded video data can be carried out with parallel readout of reference data from a plurality of memory banks based on the same motion vector, so that a plurality of reference data memory banks and motion compensation circuits can be operated in parallel to carry out high speed processing on the basis of a conventional encoding method that is not modified by limiting the range of motion vectors, or by placing other limitations on motion prediction.

As used in the specification and the following claims, the term "image frame" should be understood to mean a signal representing a picture upon which motion-compensated predictive coding is performed. As will be understood by those skilled in the art, such a picture may be formed, for example, of a progressive-scanned video frame, one field of an interlace-scanned video frame, or two fields which together make up an interlace-scanned video frame.

In at least preferred embodiments there is provided a method and apparatus for decoding a video signal in which a plurality of memory units and motion compensation devices are operated in parallel to process video data encoded according to a known standard, and without limiting the range of motion vectors used for predictive coding or requiring similar restrictions on motion predictive compression-coding.

Having described specific preferred embodiments of the present invention with reference to the accompanying drawings, it is to be understood that the invention is not limited to those precise embodiments, and that various changes and modifications may be effected by one skilled in the art without departing from the scope of the invention as defined in the appended claims.

Claims

1. An apparatus for decoding a coded video signal that represents an image frame, said coded video signal having been divided into a plurality of slices (1, 2, 3, 4), each of said slices being a sequence of macroblocks (MB), each of said macroblocks being a two-dimensional array of picture elements of said image frame, said coded video signal being a bit stream that represents a sequence of said slices which together represent said image frame, said bit stream including a plurality of synchronizing code signals, each of which is associated with a respective one of said slices for indicating a beginning of the respective slice, the apparatus comprising:

   a plurality of decoding means (30, 31, 32, 33), each for decoding a respective portion of said coded video signal that represents said image frame; and

   distributing means (25) responsive to said synchronizing code signals for distributing said slices among said plurality of decoding means.

2. An apparatus according to claim 1, wherein said plurality of decoding means is fewer in number than said plurality of slices into which said coded video signal which represents said image frame was divided, and said distributing means distributes said slices in cyclical fashion among said decoding means.

3. An apparatus according to any one of claims 1 and 2, wherein each of said slices represents a portion of said image frame which is one macroblock high and extends horizontally entirely across said image frame.

4. An apparatus according to claim 3, wherein each of said macroblocks is a 16 x 16 array of said picture elements.

5. An apparatus for decoding input signal blocks that were formed by transform encoding and then variable-length encoding blocks of video data, the apparatus comprising:

   decoding means (30, 31, 32. 33) for variable-length decoding a series of said input signal blocks;

   parallel data means (34) for forming plural parallel data streams, each of which includes respective ones of said series of input signal blocks which were variable-length decoded by said decoding means; and

   a plurality of inverse transform means (39, 40, 41, 42) each for receiving a respective one of said parallel data streams and for performing inverse transform processing on the variable-length decoded signal blocks in the respective data stream.

6. An apparatus according to claim 5, wherein said decoding means is one of a plurality of decoding means for variable-length decoding respective series of input signal blocks; and further comprising distributing means (25) for forming said respective series of input signal blocks to be decoded by said plural decoding means from a bit stream representing an image frame and in response to synchronizing signals provided at predetermined intervals in said bit stream representing said image frame.

7. An apparatus for decoding an input digital video signal which includes groups of blocks (83) of prediction-coded difference data, each of said groups consisting of a predetermined plurality of said blocks (MB) and having a respective motion vector (82) associated therewith, each of said blocks of prediction-coded difference data having been formed on the basis of the respective motion vector associated with the respective group which includes said block, the apparatus comprising:

   output means (39, 40, 41, 42) for supplying in parallel blocks of prediction-coded difference data contained in one of said groups of blocks;

   reference data means (43, 53, 54, 55, 56, 48, 49, 50, 51) for supplying in parallel plural blocks of reference data, each of said blocks of reference data being formed on the basis of the motion vector associated with said one of said groups of blocks and corresponding to one of said blocks of prediction-coded difference data supplied by said output means; and

   a plurality of adding means (57, 58, 59, 60) each connected to said output means and said

reference data means for adding a respective one of said blocks of prediction-coded difference data and the corresponding block of reference data.

8. An apparatus according to claim 7, wherein each of said groups of blocks is a macroblock which includes four blocks of prediction-coded data and said plurality of adding means consists of four adders (57, 58, 59, 60) operating in parallel.

9. An apparatus according to any one of claims 7 and 8, wherein said reference data means comprises:

a plurality of reference data memories (44, 45, 46, 47) from which reference data is read out in parallel on the basis of said motion vector associated with said one of said groups of blocks;

a plurality of buffer memories (48, 49, 50, 51), each for temporarily storing the reference data read out from a respective one of said plurality of reference data memories and for reading out the temporarily stored data on the basis of said motion vector associated with said one of said group of blocks; and

distributing means (52) connected between said buffer memories and said adding means for distributing among said plurality of adding means, on the basis of said motion vector associated with said one of said groups of blocks, the reference data read out from said plurality of buffer memories.

10. An apparatus according to any one of claims 7 and 8, wherein said reference data means comprises:

a plurality of reference data memories (44, 45, 46, 47) from which reference data is read out in parallel on the basis of said motion vector associated with said one of said groups of blocks;

a plurality of buffer memories (48, 40, 50, 51), each connected to a respective one of said adding means, for temporarily storing reference data read out from said plurality of reference data memories and for supplying the temporarily stored reference data to its respective adding means; and

distributing means (52) connected between said reference data memories and said buffer memories for distributing among the plurality of buffer memories, on the basis of said motion vector associated with said one of said groups of blocks, the reference data read out from the plurality of reference data memories.

11. An apparatus according to claim 10, wherein each of said buffer memories temporarily stores reference data read out from every one of said reference data memories.

12. An apparatus according to any one of claims 7 to 11, wherein said input digital video signal includes input signal blocks that were formed by transform encoding and then variable-length encoding blocks of prediction-coded difference data, and said output means comprises:

decoding means (30, 31, 32, 33) for variable-length decoding a series of said input signal blocks;

parallel data means (34) for forming plural parallel data streams, each of which includes respective ones of said series of input signal blocks which were variable-length decoded by said decoding means; and

a plurality of inverse transform means (39, 40, 41, 42) each for receiving a respective one of said parallel data streams and for performing inverse transform processing on the variable-length decoded signal blocks in the respective data stream to form blocks of prediction-coded difference data that are supplied to said adding means.

13. An apparatus according to claim 12, wherein said decoding means is one of a plurality of decoding means (30, 31, 32, 33) for variable-length decoding respective series of input signal blocks; and further comprising distributing means (25) for forming said respective series of input signal blocks to be decoded by said plural decoding means from a bit stream representing an image frame and in response to synchronizing signals provided at predetermined intervals in said bit stream representing said image frame.

14. A method of decoding a coded video signal that represents an image frame, said coded video signal having been divided into a plurality of slices (1, 2, 3, 4), each of said slices being a sequence of macroblocks (MB), each of said macroblocks being a two-dimensional array of picture elements of said image frame, said coded video signal being a bit stream that represents a sequence of said slices which together represent said image frame, said bit stream including a plurality of synchronizing code signals, each of which is associated with a respective one of said slices for indicating a beginning of the respective slice, the method comprising the steps of:

providing a plurality of decoding means (30, 31, 32, 33), each for decoding a respective portion of said coded signal that represents said video frame; and

distributing said slices among said plurality of decoding means in response to said synchronizing code signals.

15. A method according to claim 14, wherein said

plurality of decoding means is fewer in number than said plurality of slices into which said coded video signal which represents said image frame was divided, and said distributing step includes distributing said slices in cyclical fashion among said decoding means.

16. A method according to claim 14, wherein each of said slices represents a portion of said image frame which is one macroblock high and extends entirely across said image frame.

17. A method according to claim 16, wherein each of said macroblocks is a 16 x 16 array of said picture elements.

18. A method of decoding input signal blocks that were formed by transform encoding and then variable-length encoding blocks of video data, the method comprising the steps of:
variable-length decoding a series of said input signal blocks;
forming plural parallel data streams, each of which includes respective ones of said variable-length decoded series of input signal blocks; and
performing, in parallel, inverse transform processing on the variable-length decoded signal blocks in the respective data streams.

19. A method according to claim 18, further comprising the steps of:
forming in parallel plural series of input signal blocks from a bit stream representing an image frame of input video signals and in response to synchronizing signals provided at predetermined intervals in said bit stream representing said frame of input signals; and
variable-length decoding, in parallel, the plural series of input signal blocks.

20. A method according to claim 19, further comprising the step of distributing variable-length decoded input signal blocks from every one of said plural series of input signal blocks to each of said plural parallel data streams.

21. A method of decoding an input digital video signal which includes groups of blocks of prediction-coded difference data, each of said groups consisting of a predetermined plurality of said blocks and having a respective motion vector associated therewith, each of said blocks of prediction-coded difference data having been formed on the basis of the respective motion vector associated with the respective group which includes said block, the method comprising the steps of:
outputting in parallel blocks of prediction-coded difference data contained in one of said groups of blocks;
reading out in parallel from memory, on the basis of the motion vector associated with said one of said groups of blocks, plural blocks of reference data, each of said blocks of reference data corresponding to one of said blocks of prediction-coded difference data; and
respectively adding, in parallel, the blocks of prediction-coded difference data contained in said one of said groups of blocks and the corresponding blocks of reference data.

22. A method according to claim 21, wherein said reading out step comprises the sub-steps of:
reading out the reference data from a plurality of memories on the basis of the motion vector associated with said one of said groups of blocks;
distributing, on the basis of the motion vector associated with said one of said groups of blocks, the reference data read out from the plurality of memories;
temporarily storing the distributed reference data; and
reading out the temporarily stored data.

23. A method according to any one of claims 21 and 22, wherein said input digital video signal includes input signal blocks that were formed by transform-encoding and then variable-length encoding blocks of prediction-coded difference data, said outputting step comprising the substeps of:
variable length decoding a series of said input signal blocks;
forming plural parallel data streams, each of which includes respective ones of said variable-length decoded series of input signal blocks; and
performing, in parallel, inverse transform processing on the variable-length decoded signal blocks in the respective data streams.

24. A method according to claim 23, further comprising the steps of:
forming in parallel plural series of input signal blocks from a bit stream representing an image frame of input video signals and in response to synchronizing signals provided at predetermined intervals in said bit stream representing said frame of input signals; and
variable-length decoding, in parallel, the plural series of input signal blocks.

25. A method of decoding a prediction-coded video signal that represents an image frame, said prediction-coded video signal having been divided

into a plurality of macroblocks, each of said macroblocks being a two-dimensional array of picture elements of said image frame, the method comprising the steps of:

providing a plurality of memories each for storing reference data which corresponds to a respective portion of said image frame, said plurality of memories together storing reference data which represents a complete image frame; and

distributing data representing a reconstructed image frame for storage in said plurality of memories such that a portion of each macroblock of the reconstructed image frame is stored in each of said plurality of memories.

26. A method according to claim 25, wherein said macroblocks are each composed of a predetermined number of two-dimensional blocks and each of said plurality of memories stores corresponding blocks from all of the macroblocks of an image frame.

27. A method according to claim 26, wherein said plurality of memories consists of first, second, third and fourth memories, said macroblocks are each composed of four blocks which respectively represent upper left, upper right, lower left and lower right quadrants of the respective macroblock, and said distributing step comprises:

storing in the first memory the blocks representing the upper left quadrants of all of the macroblocks;

storing in the second memory the blocks representing the upper right quadrants of all of the macroblocks;

storing in the third memory the blocks representing the lower left quadrants of all of the macroblocks; and

storing in the fourth memory the blocks representing the lower right quadrants of all of the macroblocks.

28. A method according to any one of claims 25, 26 and 27, wherein said distributing step comprises storing a first line of each of said macroblocks in a first one of said plurality of memories and storing a second line of each of said macroblocks in a second one of said plurality of memories.

29. A method according to claim 28, wherein each of said macroblocks is composed of a number of lines that is an integral multiple of a number of memories that forms said plurality of memories, and said distributing step comprises distributing said lines of each macroblock in cyclical fashion among said memories.

30. A method according to claim 29, wherein each of said macroblocks is composed of sixteen lines, and said number of memories is four.

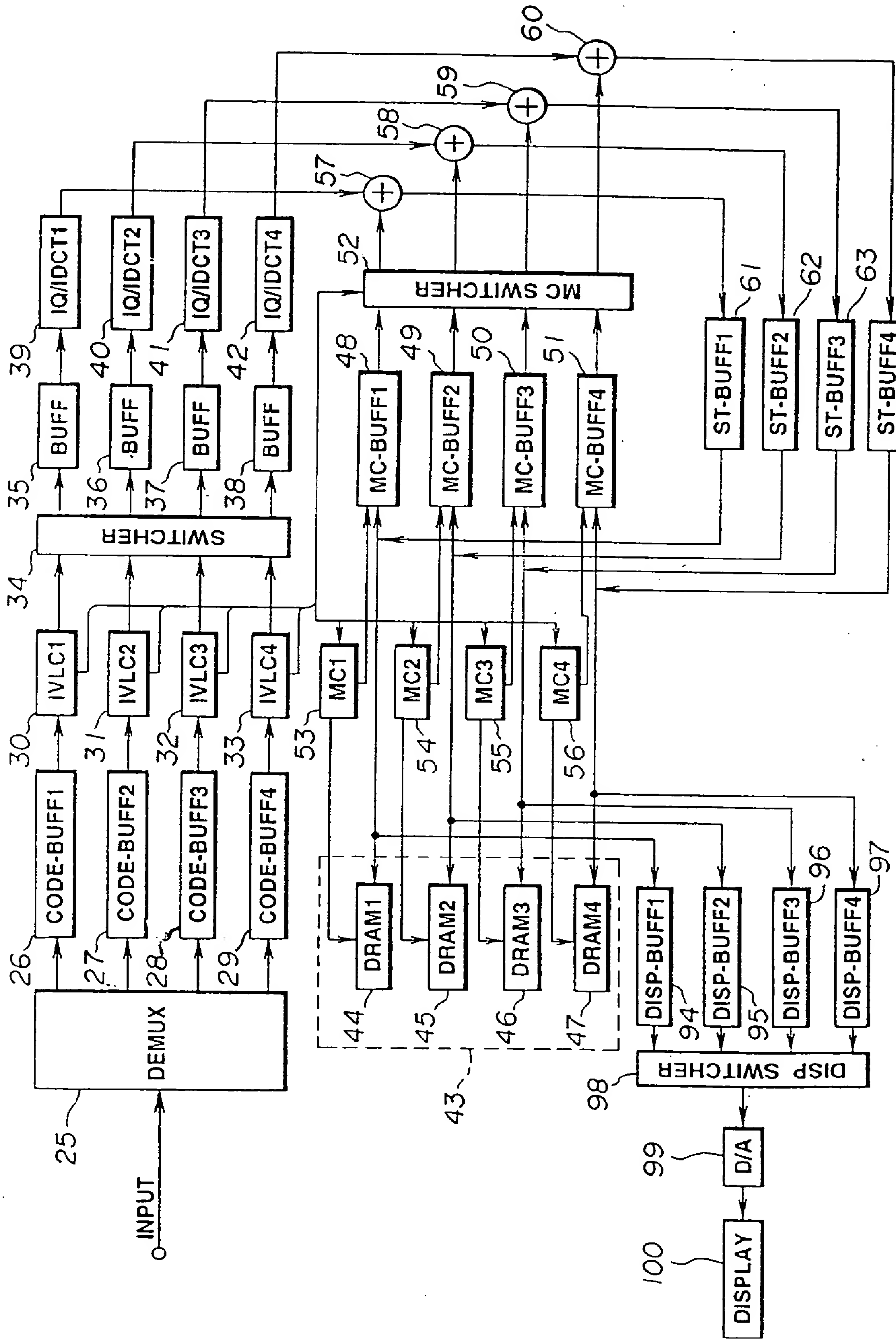
INPUT
DEMUX
CODE-BUFF1
CODE-BUFF2
CODE-BUFF3
CODE-BUFF4
IVLC1
IVLC2
IVLC3
IVLC4
SWITCHER
BUFF
BUFF
BUFF
BUFF
IQ/IDCT1
IQ/IDCT2
IQ/IDCT3
IQ/IDCT4
MC1
MC2
MC3
MC4
MC-BUFF1
MC-BUFF2
MC-BUFF3
MC-BUFF4
MC SWITCHER
DRAM1
DRAM2
DRAM3
DRAM4
ST-BUFF1
ST-BUFF2
ST-BUFF3
ST-BUFF4
DISP-BUFF1
DISP-BUFF2
DISP-BUFF3
DISP-BUFF4
DISP SWITCHER
D/A
DISPLAY

FIG.1

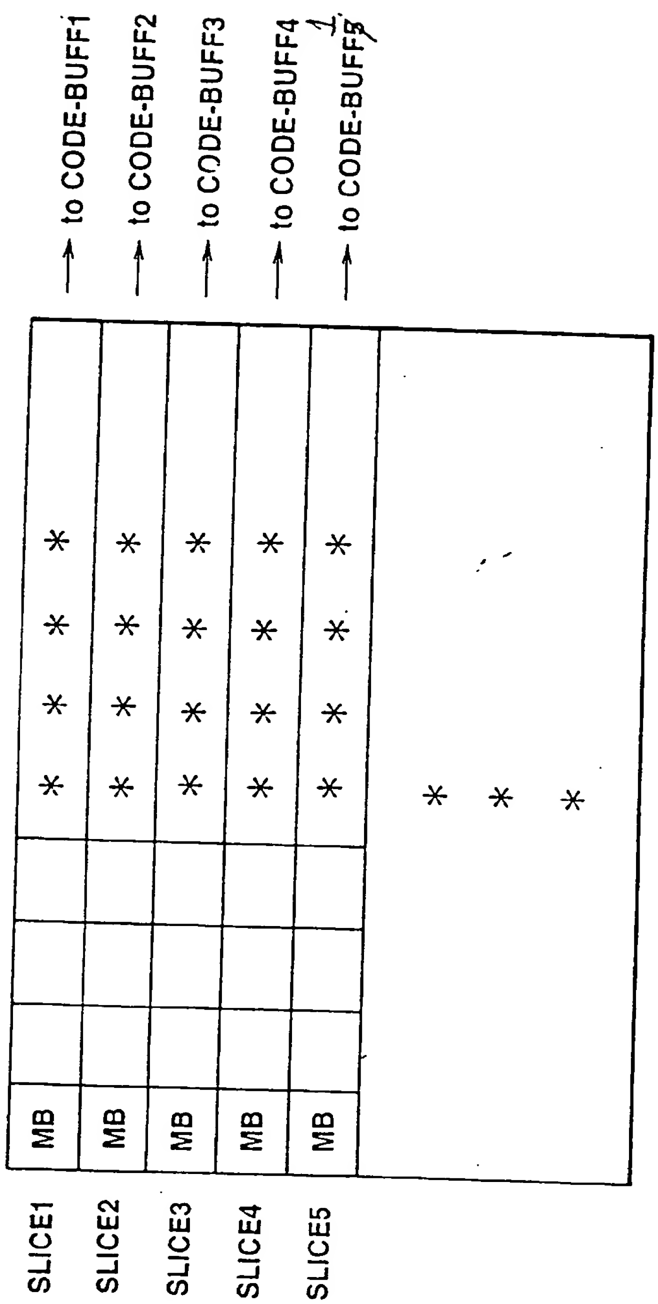
to CODE-BUFF1
to CODE-BUFF2
to CODE-BUFF3
to CODE-BUFF4
to CODE-BUFF5
MB
MB
MB
MB
MB
SLICE1
SLICE2
SLICE3
SLICE4
SLICE5

FIG.2

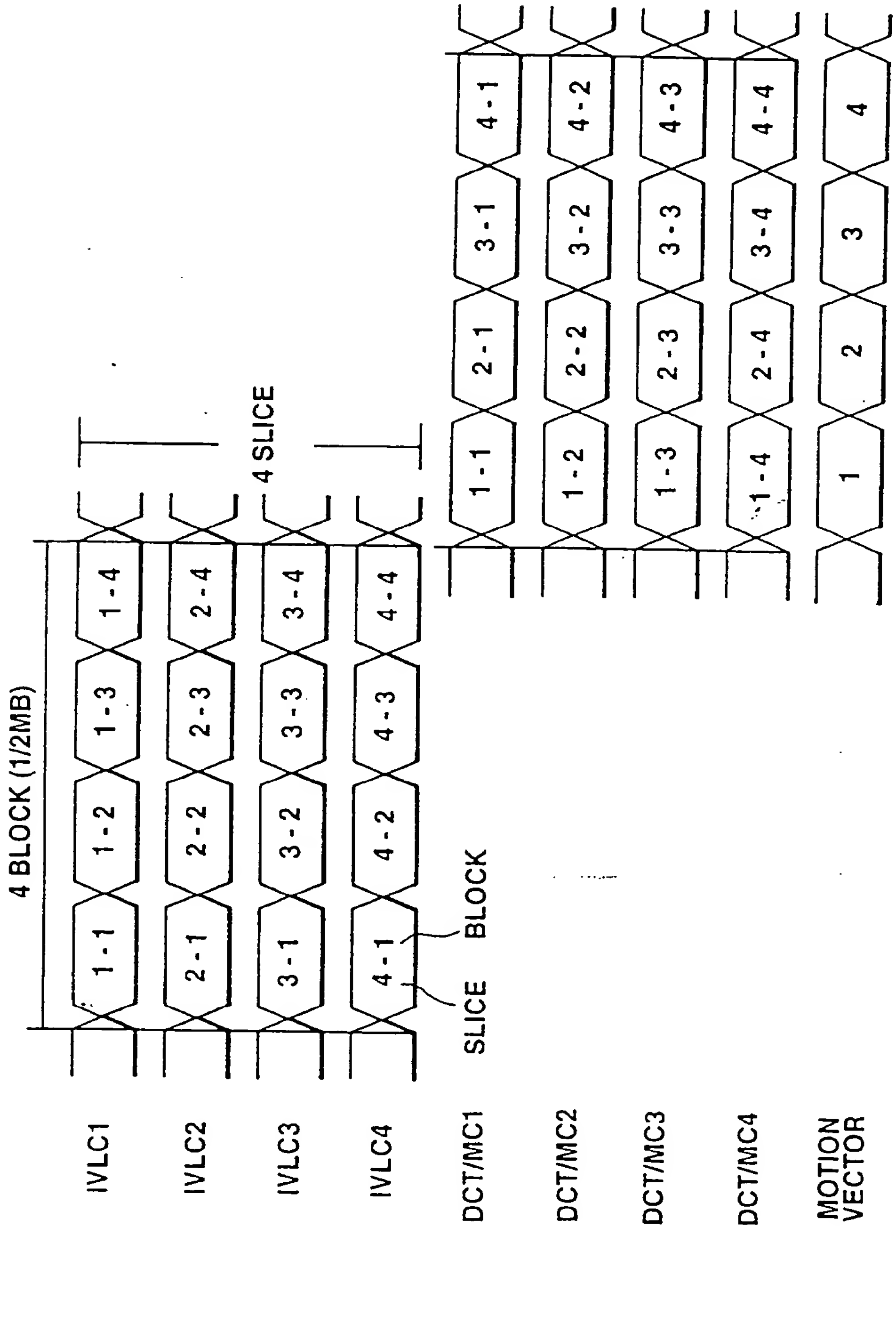
4 BLOCK (1/2MB)
4 SLICE
IVLC1
IVLC2
IVLC3
IVLC4
DCT/MC1
DCT/MC2
DCT/MC3
DCT/MC4
MOTION VECTOR
SLICE
BLOCK
1 - 1
1 - 2
1 - 3
1 - 4
2 - 1
2 - 2
2 - 3
2 - 4
3 - 1
3 - 2
3 - 3
3 - 4
4 - 1
4 - 2
4 - 3
4 - 4
1
2
3
4

FIG.3

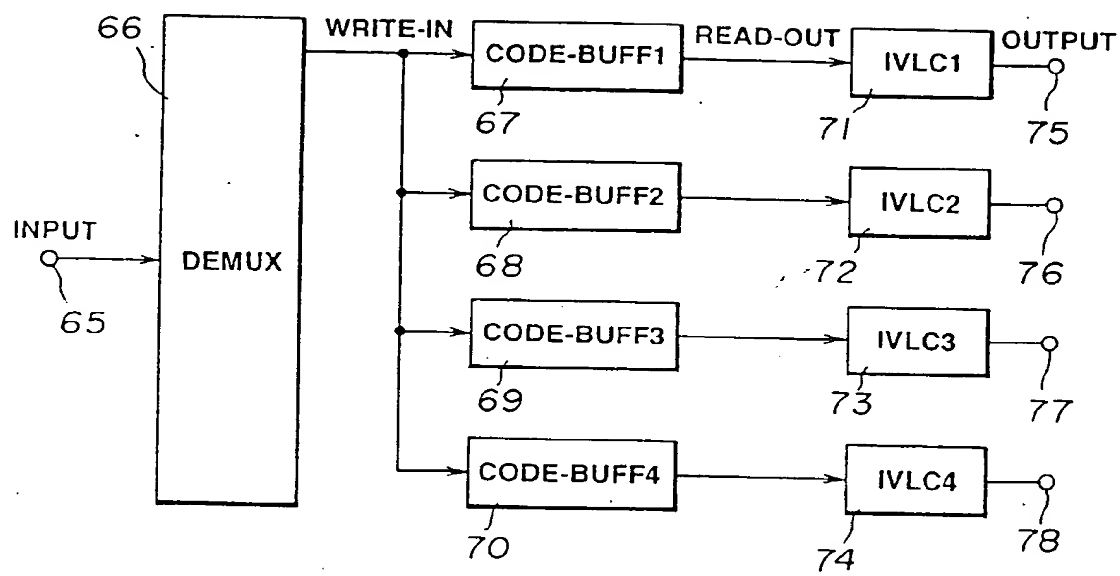
66
WRITE-IN
CODE-BUFF1
READ-OUT
IVLC1
OUTPUT
67
71
75
CODE-BUFF2
IVLC2
INPUT
DEMUX
68
72
76
65
CODE-BUFF3
IVLC3
69
73
77
CODE-BUFF4
IVLC4
70
74
78

FIG.4

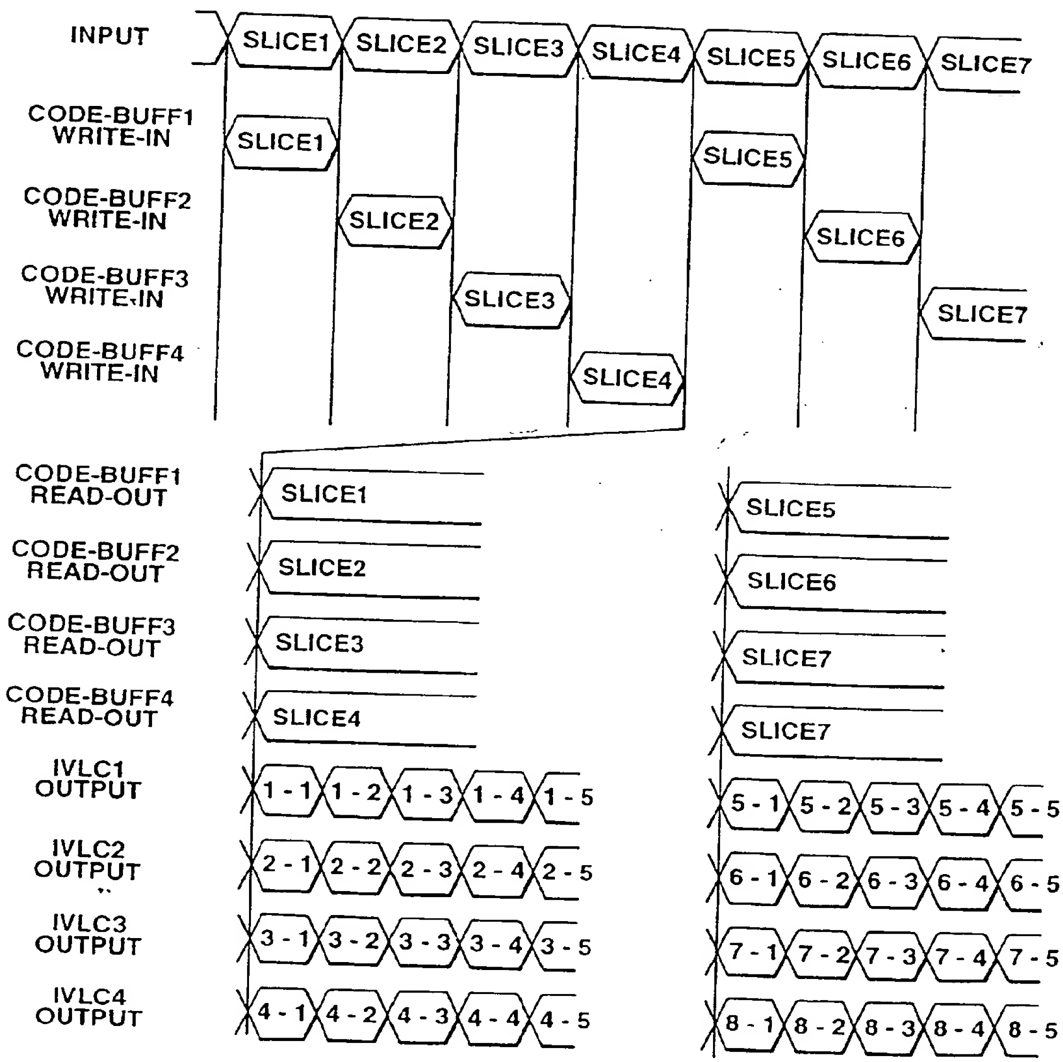
INPUT
SLICE1
SLICE2
SLICE3
SLICE4
SLICE5
SLICE6
SLICE7
CODE-BUFF1 WRITE-IN
SLICE1
SLICE5
CODE-BUFF2 WRITE-IN
SLICE2
SLICE6
CODE-BUFF3 WRITE-IN
SLICE3
SLICE7
CODE-BUFF4 WRITE-IN
SLICE4
CODE-BUFF1 READ-OUT
SLICE1
SLICE5
CODE-BUFF2 READ-OUT
SLICE2
SLICE6
CODE-BUFF3 READ-OUT
SLICE3
SLICE7
CODE-BUFF4 READ-OUT
SLICE4
SLICE7
IVLC1 OUTPUT
1 - 1
1 - 2
1 - 3
1 - 4
1 - 5
5 - 1
5 - 2
5 - 3
5 - 4
5 - 5
IVLC2 OUTPUT
2 - 1
2 - 2
2 - 3
2 - 4
2 - 5
6 - 1
6 - 2
6 - 3
6 - 4
6 - 5
IVLC3 OUTPUT
3 - 1
3 - 2
3 - 3
3 - 4
3 - 5
7 - 1
7 - 2
7 - 3
7 - 4
7 - 5
IVLC4 OUTPUT
4 - 1
4 - 2
4 - 3
4 - 4
4 - 5
8 - 1
8 - 2
8 - 3
8 - 4
8 - 5

FIG.5

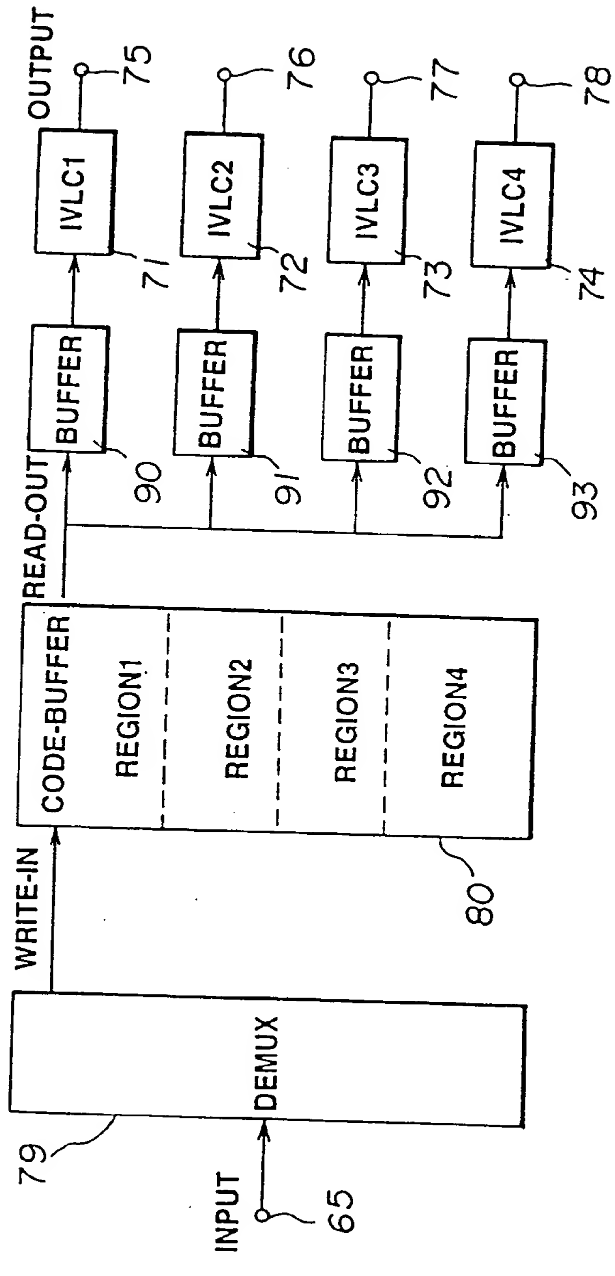
INPUT
65
79
DEMUX
WRITE-IN
CODE-BUFFER
REGION1
REGION2
REGION3
REGION4
80
READ-OUT
BUFFER
90
BUFFER
91
BUFFER
92
BUFFER
93
IVLC1
71
IVLC2
72
IVLC3
73
IVLC4
74
OUTPUT
75
76
77
78

FIG.6

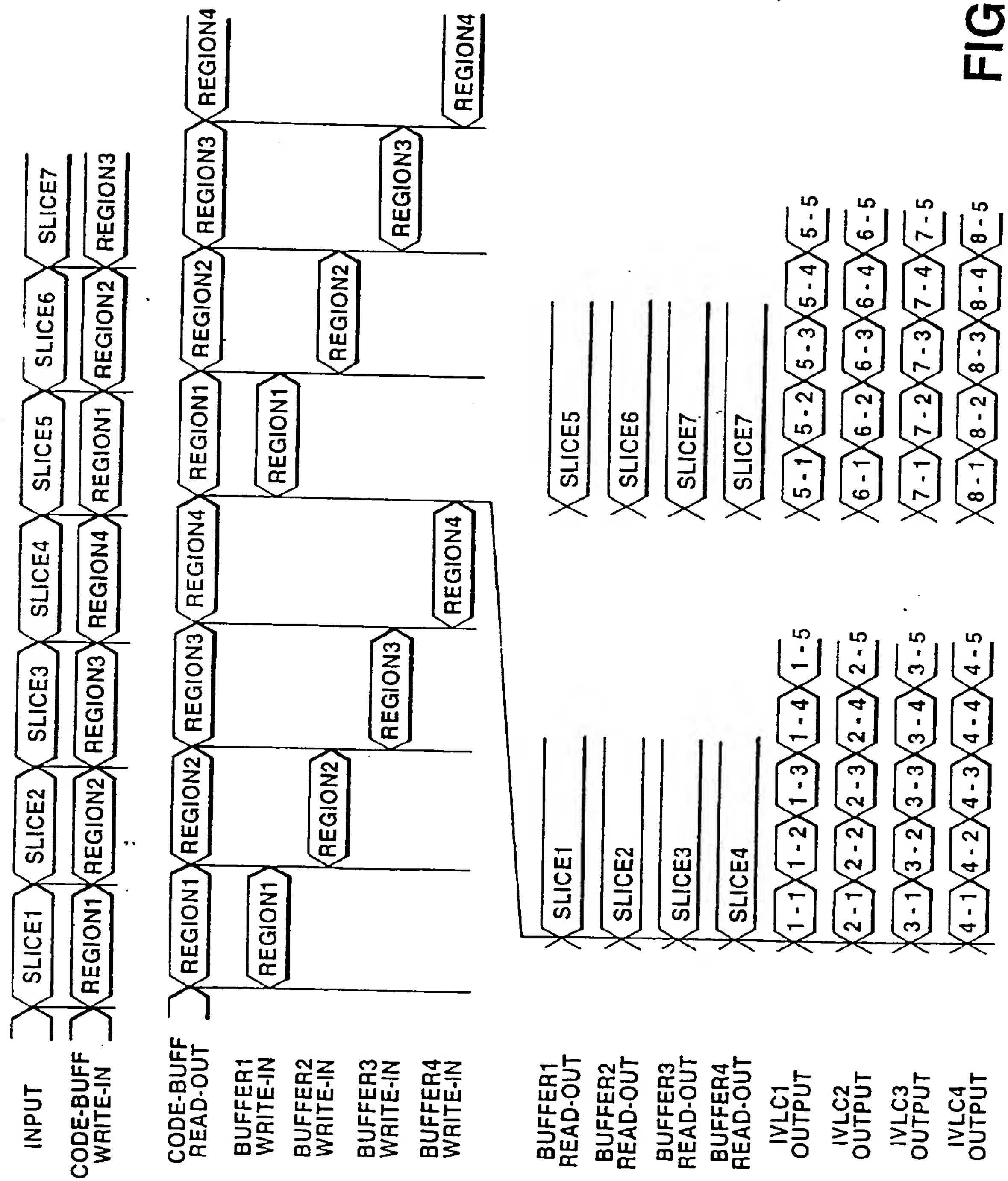
INPUT
CODE-BUFF WRITE-IN
CODE-BUFF READ-OUT
BUFFER1 WRITE-IN
BUFFER2 WRITE-IN
BUFFER3 WRITE-IN
BUFFER4 WRITE-IN
BUFFER1 READ-OUT
BUFFER2 READ-OUT
BUFFER3 READ-OUT
BUFFER4 READ-OUT
IVLC1 OUTPUT
IVLC2 OUTPUT
IVLC3 OUTPUT
IVLC4 OUTPUT
SLICE1
SLICE2
SLICE3
SLICE4
SLICE5
SLICE6
SLICE7
REGION1
REGION2
REGION3
REGION4
1-1 1-2 1-3 1-4 1-5
2-1 2-2 2-3 2-4 2-5
3-1 3-2 3-3 3-4 3-5
4-1 4-2 4-3 4-4 4-5
5-1 5-2 5-3 5-4 5-5
6-1 6-2 6-3 6-4 6-5
7-1 7-2 7-3 7-4 7-5
8-1 8-2 8-3 8-4 8-5

FIG.7

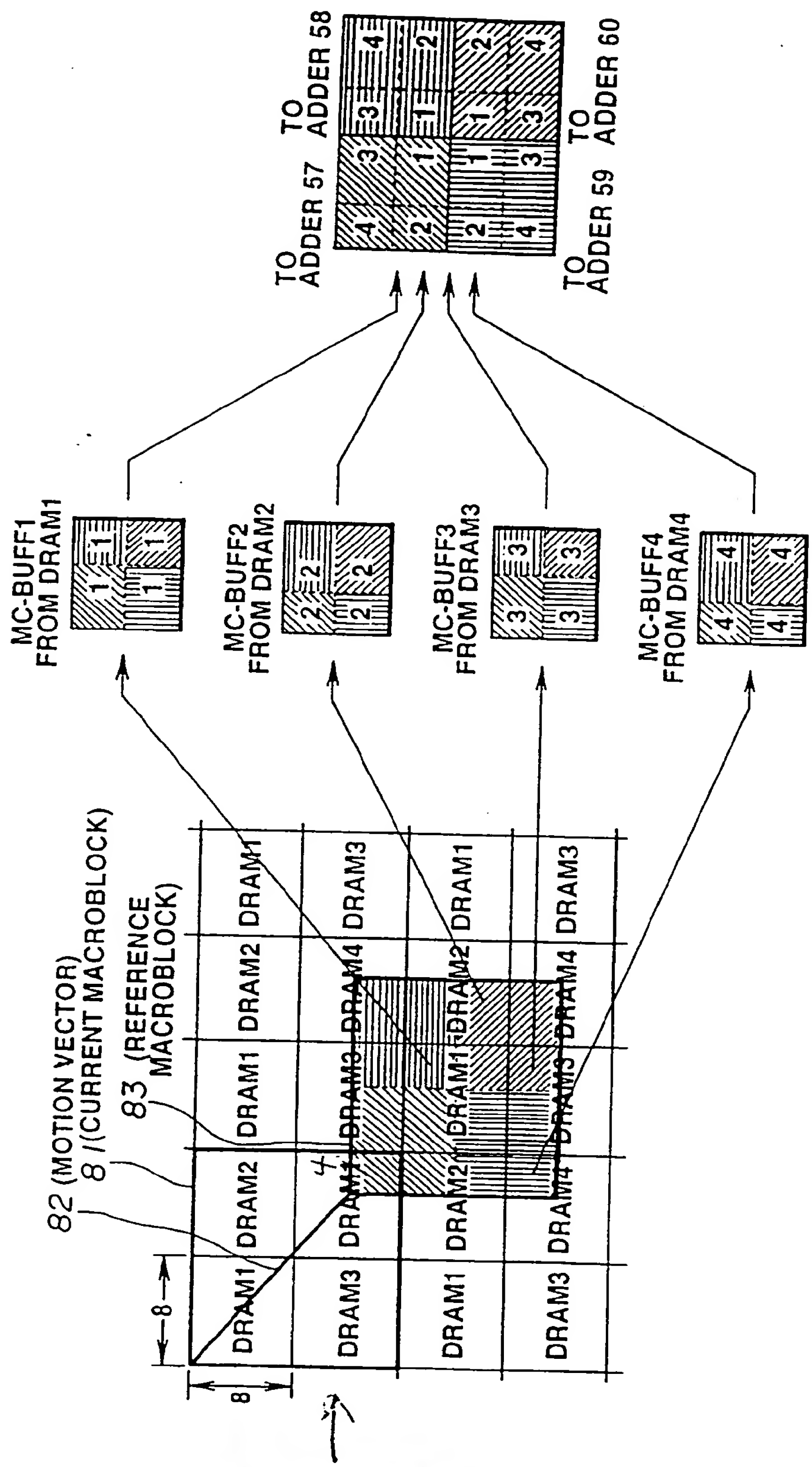

FIG.8(A)

FIG.8(B)

FIG.8(C)

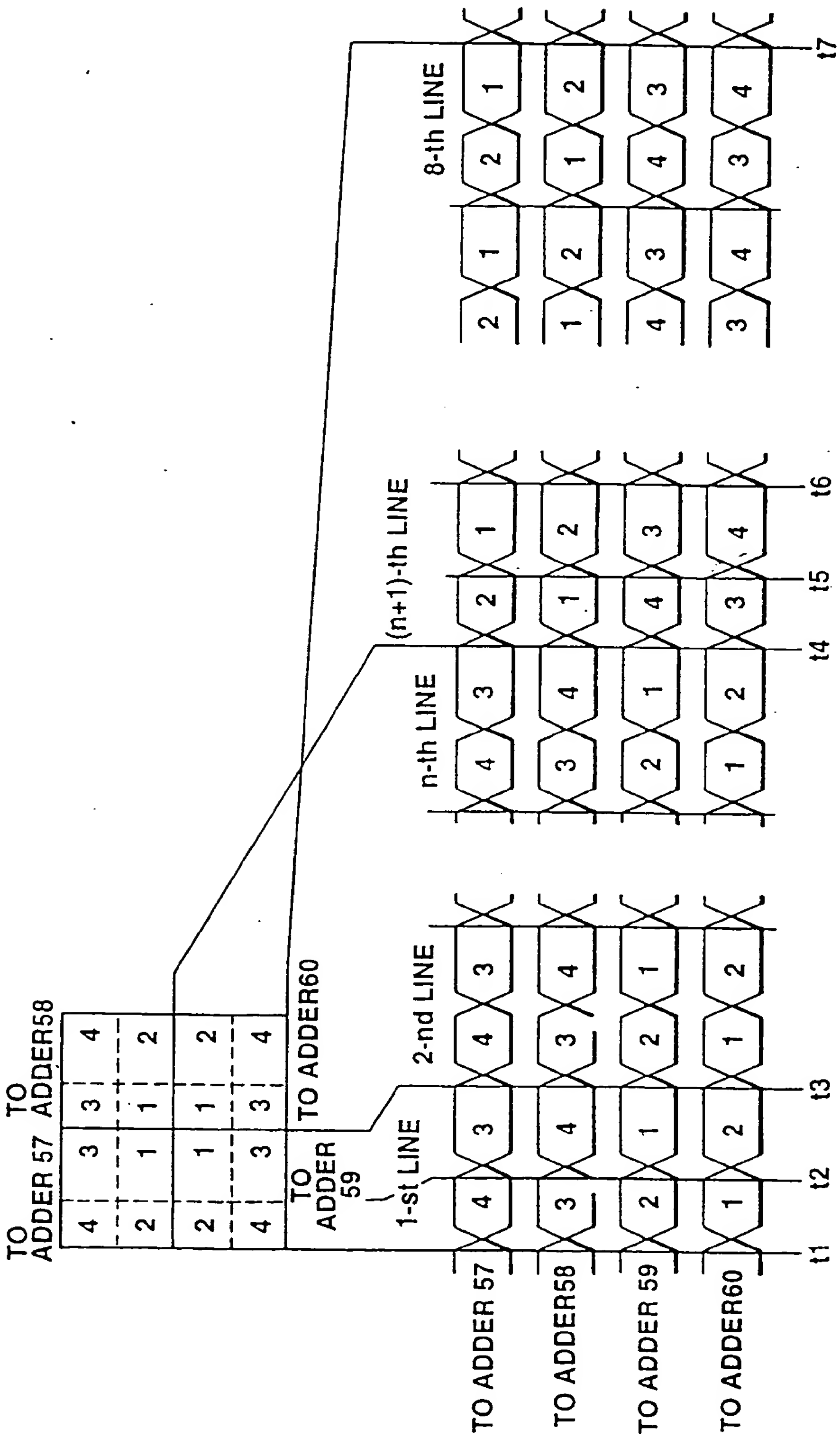
TO ADDER 57
TO ADDER58
TO ADDER 59
TO ADDER60
1-st LINE
2-nd LINE
n-th LINE
(n+1)-th LINE
8-th LINE
t1
t2
t3
t4
t5
t6
t7

FIG.9

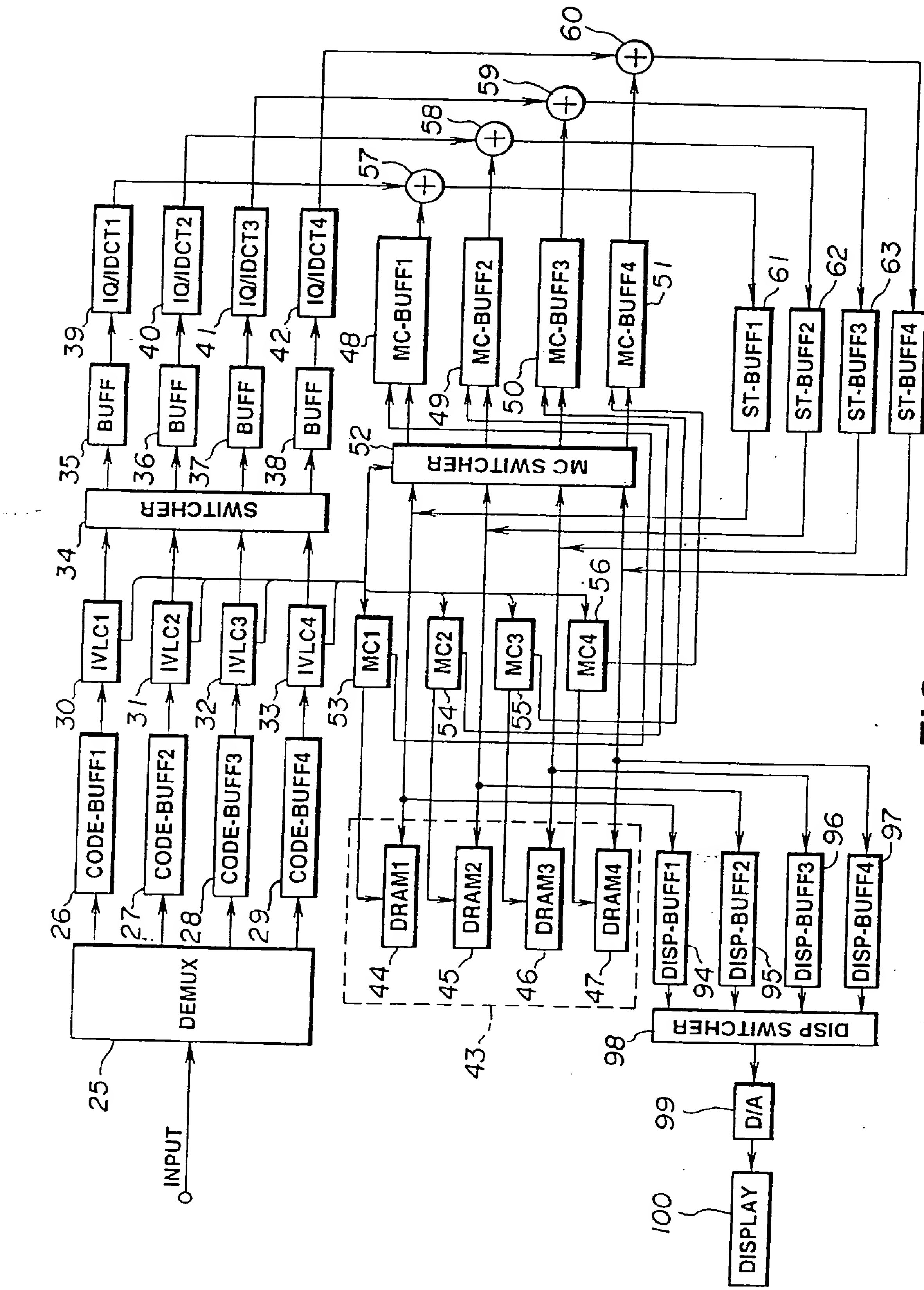
INPUT
DEMUX
CODE-BUFF1
CODE-BUFF2
CODE-BUFF3
CODE-BUFF4
IVLC1
IVLC2
IVLC3
IVLC4
SWITCHER
BUFF
BUFF
BUFF
BUFF
IQ/IDCT1
IQ/IDCT2
IQ/IDCT3
IQ/IDCT4
MC1
MC2
MC3
MC4
MC SWITCHER
MC-BUFF1
MC-BUFF2
MC-BUFF3
MC-BUFF4
DRAM1
DRAM2
DRAM3
DRAM4
ST-BUFF1
ST-BUFF2
ST-BUFF3
ST-BUFF4
DISP-BUFF1
DISP-BUFF2
DISP-BUFF3
DISP-BUFF4
DISP SWITCHER
D/A
DISPLAY
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
94
95
96
97
98
99
100

FIG.10

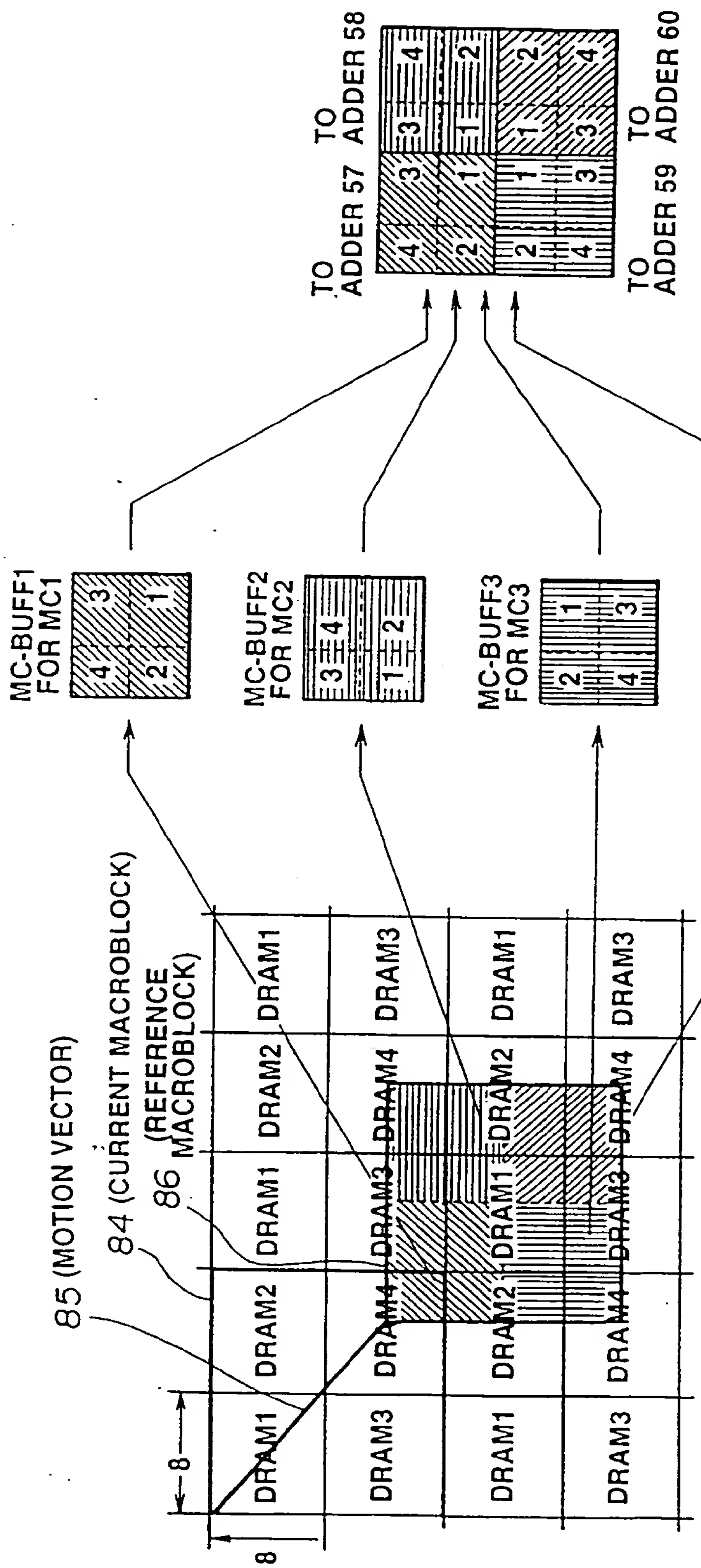

FIG.11(A)

FIG.11(B)

FIG.11(~~B~~C)

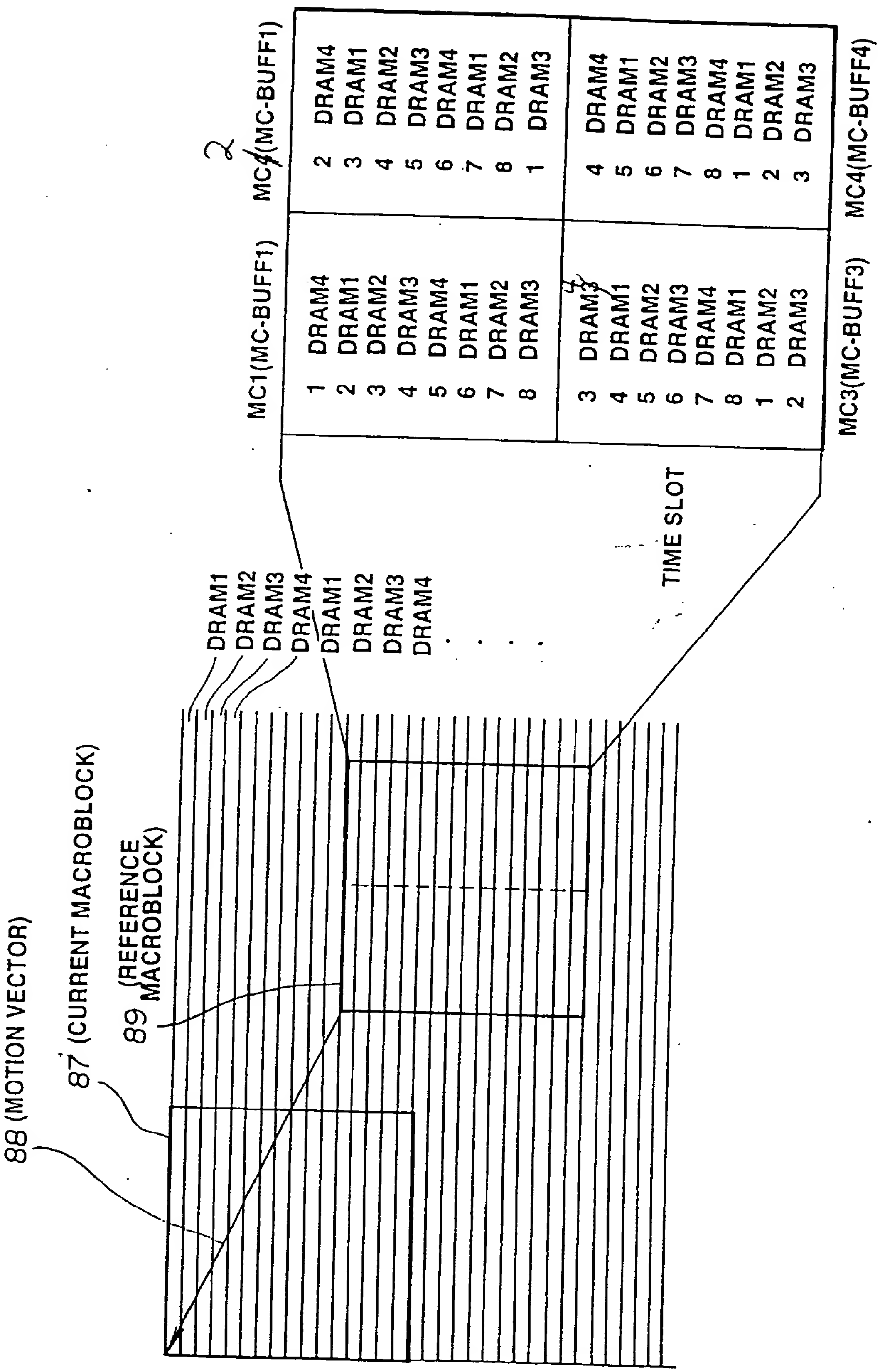

FIG.12(A)

FIG.12(B)

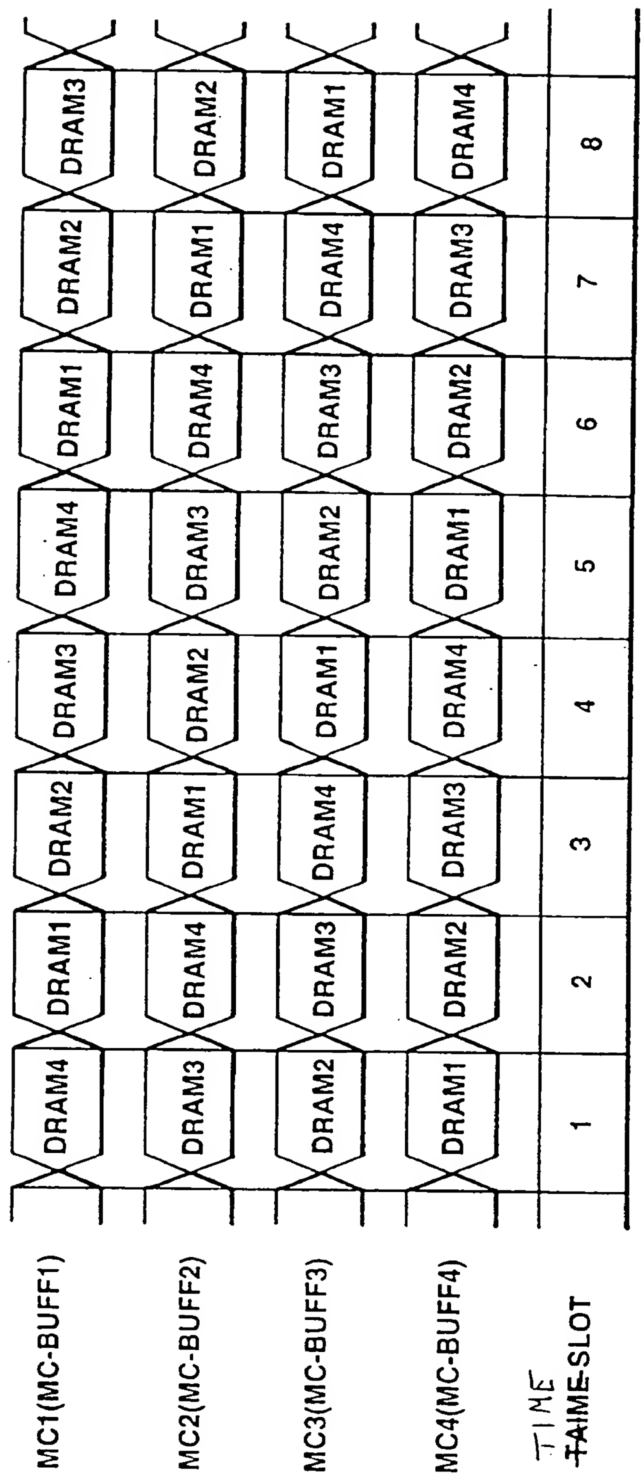
MC1(MC-BUFF1)
DRAM4 DRAM1 DRAM2 DRAM3 DRAM4 DRAM1 DRAM2 DRAM3
MC2(MC-BUFF2)
DRAM3 DRAM4 DRAM1 DRAM2 DRAM3 DRAM4 DRAM1 DRAM2
MC3(MC-BUFF3)
DRAM2 DRAM3 DRAM4 DRAM1 DRAM2 DRAM3 DRAM4 DRAM1
MC4(MC-BUFF4)
DRAM1 DRAM2 DRAM3 DRAM4 DRAM1 DRAM2 DRAM3 DRAM4
TIME
TIME-SLOT
1 2 3 4 5 6 7 8

FIG.13

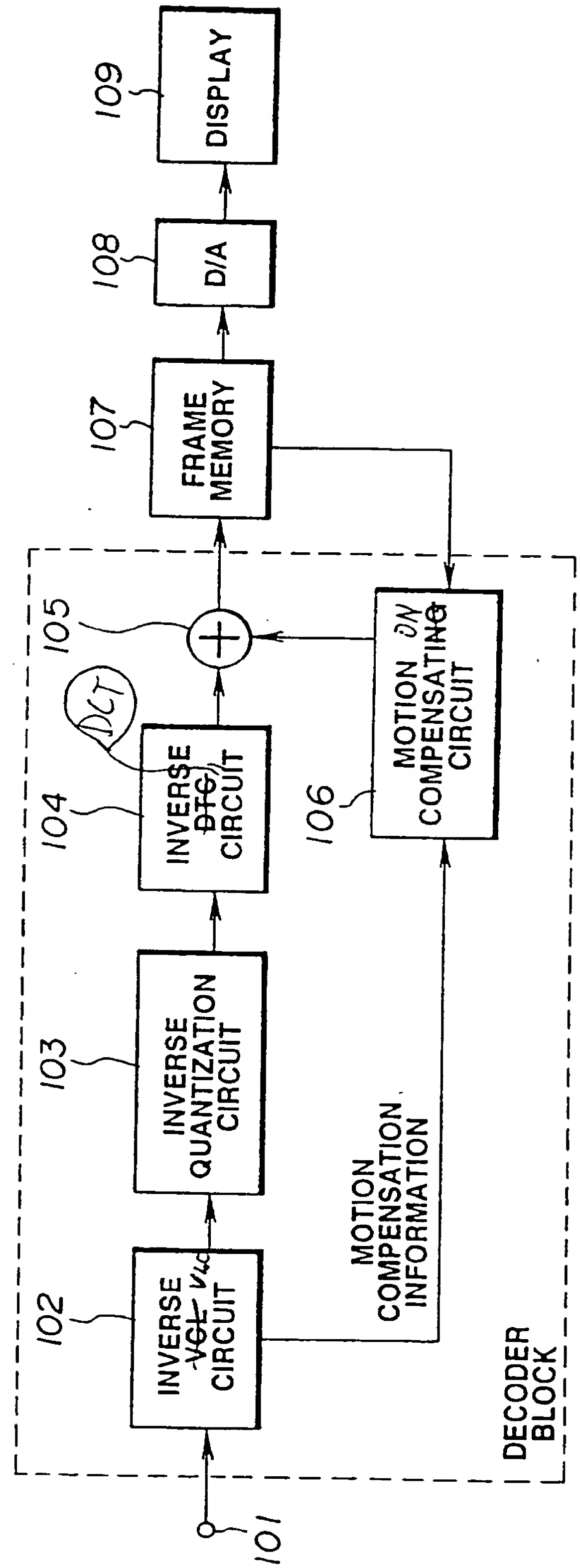
101
102
INVERSE ~~VCL~~ VLC CIRCUIT
103
INVERSE QUANTIZATION CIRCUIT
104
INVERSE ~~DTC~~ CIRCUIT
DCT
105
106
MOTION ~~COMPENSATING~~ COMPENSATION CIRCUIT
107
FRAME MEMORY
108
D/A
109
DISPLAY
MOTION COMPENSATION INFORMATION
DECODER BLOCK

FIG.14

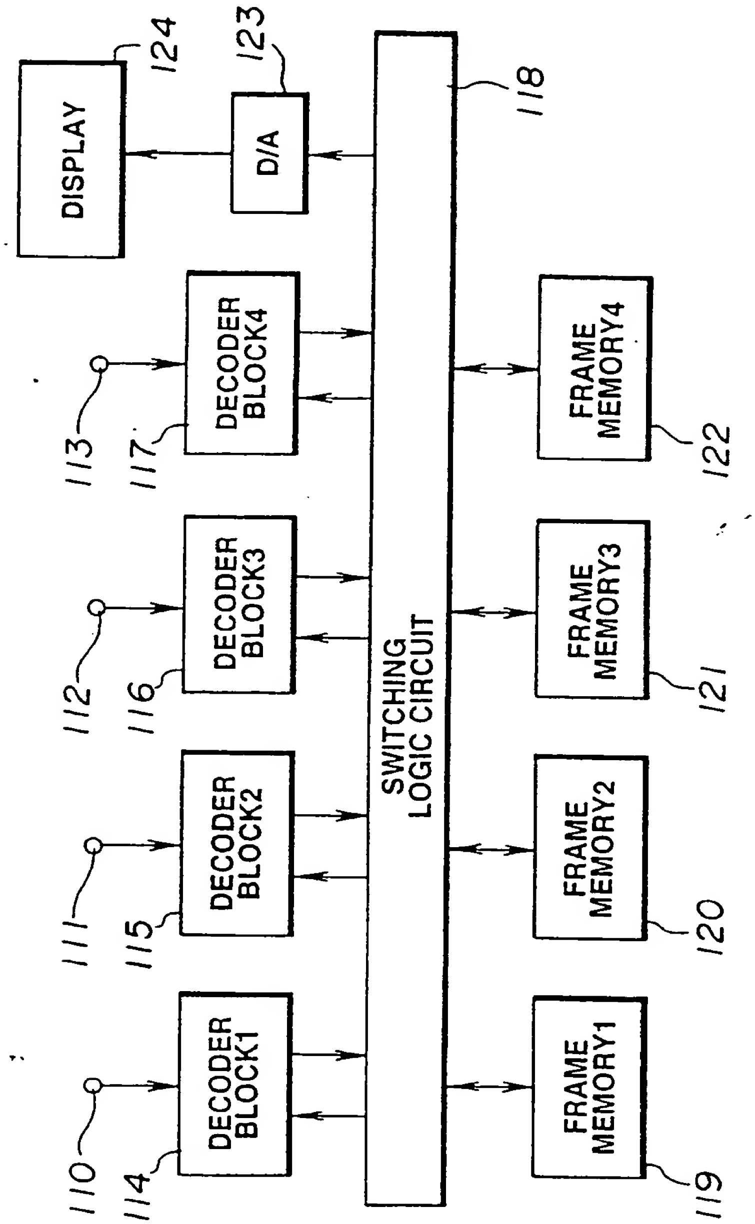
DISPLAY
D/A
SWITCHING LOGIC CIRCUIT
DECODER BLOCK1
DECODER BLOCK2
DECODER BLOCK3
DECODER BLOCK4
FRAME MEMORY1
FRAME MEMORY2
FRAME MEMORY3
FRAME MEMORY4
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124

FIG.15

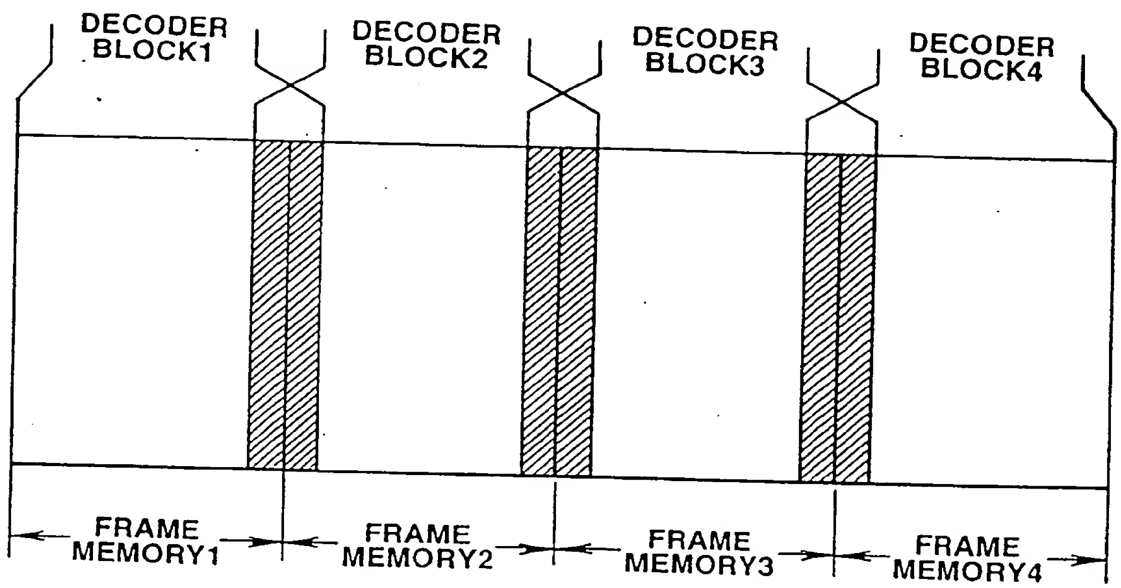
DECODER BLOCK1
DECODER BLOCK2
DECODER BLOCK3
DECODER BLOCK4
FRAME MEMORY1
FRAME MEMORY2
FRAME MEMORY3
FRAME MEMORY4

FIG.16

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : 0 614 317 A3

(12) EUROPEAN PATENT APPLICATION

(21) Application number : **94301252.6**

(22) Date of filing : **22.02.94**

(51) Int. Cl.^{5} : **H04N 7/13,** H04N 7/137

(30) Priority : **05.03.93 JP 45112/93**

(43) Date of publication of application : **07.09.94 Bulletin 94/36**

(84) Designated Contracting States : **DE FR GB IT NL**

(88) Date of deferred publication of search report : **25.01.95 Bulletin 95/04**

(71) Applicant : **SONY CORPORATION**
7-35 Kitashinagawa 6-chome
Shinagawa-ku
Tokyo 141 (JP)

(72) Inventor : **Koyanagi, Hideki, c/o Intellectual Property Div.**
Sony Corporation,
6-7-35 Kitashinagawa
Shinagawa-ku, Tokyo 141 (JP)
Inventor : **Sumihiro, Hiroshi, c/o Intellectual Property Div.**
Sony Corporation,
6-7-35 Kitashinagawa
Shinagawa-ku, Tokyo 141 (JP)
Inventor : **Emoto, Seiichi, c/o Intellectual Property Div.**
Sony Corporation,
6-7-35 Kitashinagawa
Shinagawa-ku, Tokyo 141 (JP)
Inventor : **Wada, Tohru, c/o Intellectual Property Div.**
Sony Corporation,
6-7-35 Kitashinagawa
Shinagawa-ku, Tokyo 141 (JP)

(74) Representative : **Robinson, Nigel Alexander Julian et al**
D. Young & Co.,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) Video signal decoding.

(57) A digital video signal that has been encoded using motion- compensated prediction, transform encoding, and variable-length coding, is decoded using parallel processing. Frames of the video signal are divided into slices (1, 2, 3, 4) made up of a sequence of macroblocks (MB). The signal to be decoded is slice-wise divided for parallel variable-length decoding. Each variable-length-decoded macroblock is divided into its constituent blocks for parallel inverse transform processing. Resulting blocks of difference data are added in parallel to corresponding blocks of reference data. The blocks of reference data corresponding to each macroblock are read out in parallel from reference data memories (44, 45, 46, 47) on the basis of a motion vector (83) associated with the macroblock. Reference data corresponding to each macroblock is distributed for storage among a number of reference data memories.

EP 0 614 317 A3

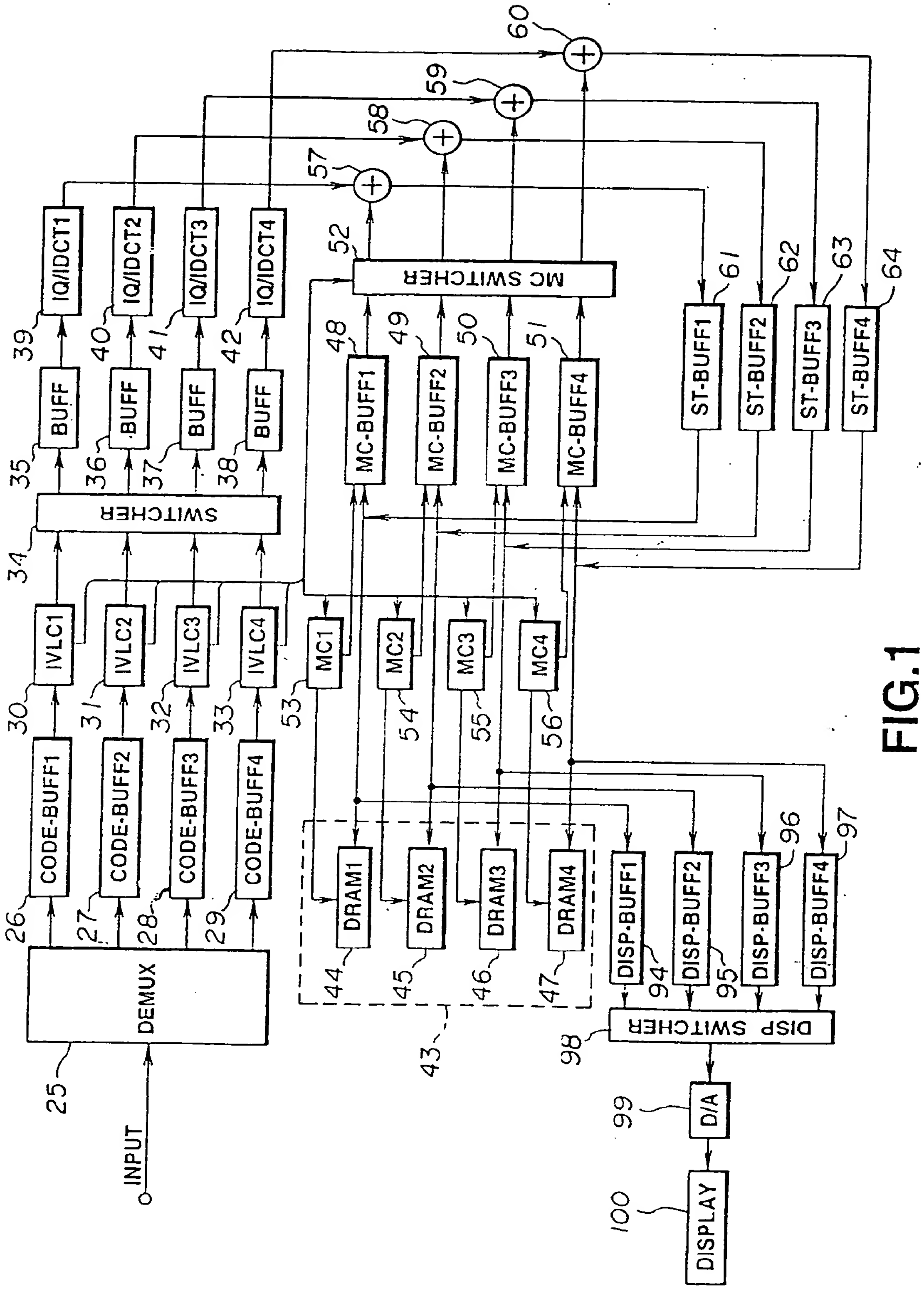
INPUT
DEMUX
CODE-BUFF1
CODE-BUFF2
CODE-BUFF3
CODE-BUFF4
IVLC1
IVLC2
IVLC3
IVLC4
SWITCHER
BUFF
BUFF
BUFF
BUFF
IQ/IDCT1
IQ/IDCT2
IQ/IDCT3
IQ/IDCT4
MC1
MC2
MC3
MC4
MC-BUFF1
MC-BUFF2
MC-BUFF3
MC-BUFF4
MC SWITCHER
ST-BUFF1
ST-BUFF2
ST-BUFF3
ST-BUFF4
DRAM1
DRAM2
DRAM3
DRAM4
DISP-BUFF1
DISP-BUFF2
DISP-BUFF3
DISP-BUFF4
DISP SWITCHER
D/A
DISPLAY
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
94
95
96
97
98
99
100

FIG.1

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 94 30 1252

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.5)
X	EBU TECHNICAL REVIEW, no.251, 1 April 1992, BRUSSEL, BE pages 22 - 33, XP275388 M. BARBERO ET AL. 'DCT source coding and current implementations for HDTV'	1-4, 14-20	H04N7/13 H04N7/137
Y	* paragraph 3.2-3.3; figures 3-5 *	5,6	
A		7-13, 21-30	
Y	EP-A-0 442 548 (LABORATOIRES D'ELECTRONIQUE PHILIPS ET AL.)	5,6	
A	* the whole document *	1-4,7-30	
A	PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON APPLICATION SPECIFIC ARRAY PROCESSORS, 5 September 1990, PRINCETON, NEW JERSEY, US pages 226 - 234, XP245104 T. NISHITANI ET AL. 'A Real-Time Software Programmable Processor for HDTV and Stereo Scope Signals' * paragraph 3; figure 3 *	1-30	TECHNICAL FIELDS SEARCHED (Int.Cl.5)
X	EP-A-0 479 511 (VICTOR COMPANY OF JAPAN, LTD.)	25,26	H04N
A	* the whole document *	1-24, 27-30	
X	IEEE MICRO, vol.12, no.5, 1 October 1992, LOS ALAMITOS, CA, US pages 22 - 27, XP330853 O. DUARDO ET AL. 'Architecture and Implementation of ICs for a DSC-HDTV Video Decoder System'	1-4, 14-20	
A	* paragraph 1 -paragraph 2; figure 4 *	5-13, 21-30	

-/--

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	16 December 1994	Foglia, P

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 94 30 1252

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.5)
P,A	US-A-5 212 742 (NORMILE ET AL.) * column 9, line 22 - column 14, line 11; figures 4-6 * ---	1-30	
X	RTM RUNDFUNKTECHNISCHE MITTEILUNGEN, vol.36, no.5, 1 September 1992, NORDERSTEDT, DE pages 196 - 205, XP321766 H. HOFMANN ET AL. 'Ein Codec für die digitale Übertragung von HDTV-Signalen' * paragraph 5; figures 6A-8 * ---	1,14	
P,A	EP-A-0 535 746 (PHILIPS ELECTRONICS UK LTD. ET AL.) ---		
A	TRANSPUTER/OCCAM JAPAN 3, 17 May 1990, TOKYO,JP pages 91 - 100 M. ICHIKAWA ET AL. 'Design and implementation of software based real-time video codec using multi-transputer architecture' * paragraph 3.1 -paragraph 4.3.5 * ---	1-30	TECHNICAL FIELDS SEARCHED (Int.Cl.5)
A	US-A-5 130 797 (MURAKAMI ET AL.) -----		

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	16 December 1994	Foglia, P

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)